Traitement du Signal
Vol. 42, No. 4, August, 2025, pp. 1841-1862
Journal homepage: http://iieta.org/journals/ts

Real-Time and Accurate Pupil Detection Based Retro-Oriented Mind and Ellipse Trend Analysis

Şerif İnanır[1,2*], Ali Can Karaca[1,3]

[1] Department of Computer Engineering, Yildiz Technical University, Istanbul 34220, Turkey
[2] Department of Software Development, Istanbul Bilgi University, Istanbul 34060, Turkey
[3] MOSAIC Research Group, Yildiz Technical University, Istanbul 34220, Turkey

Corresponding Author Email: sheriffnnr@gmail.com

https://doi.org/10.18280/ts.420402

Received: 4 December 2024
Revised: 12 April 2025
Accepted: 12 June 2025
Available online: 14 August 2025

Keywords:
pupil detection, eye tracking systems, image segmentation, real-time image processing

ABSTRACT

This study focuses on designing a pupil ellipse detector for wearable eye trackers. The detector uses both a traditional method producing pupil patches in different resolutions and a learning model segmenting these patches. Therefore, the frequency is increased as the input size of the learning model will be reduced according to the structure of the received image. This novel approach in the pupil detection field was named as Retro-Oriented Mind (ROM). The study also presents metrics measuring the segmentation accuracy and a correction mechanism improving ellipse parameters if metric scores are not acceptable. The combination of novel metrics and correction mechanisms was named as Pupil Ellipse Trend Analysis (PETA). Using ROM and PETA, the proposed study has achieved an accuracy of over 90% and a frequency of more than 120 Hz (from about 30 Hz) in analyses of LPW and Dikablis datasets. These measurements reveal the potential of the study to be used for both medical and general purposes. Code and details: https://github.com/Serif-NNR/rom-peta-pupil-detection.

1. INTRODUCTION

Eye tracking systems serve two primary purposes: (i) detecting the subject's gaze in the external environment, and (ii) understanding the effects of the area the subject observes on oneself. Both functions render eye tracking applicable across various domains, which can be categorized into three main areas: subject analysis, object analysis, and device control. Subject analysis provides valuable insights into the user's biological, psychological, educational, and experiential contexts whereas object analysis allows to evaluate the object with which the subject interacts. On the other hand, device control enables triggering and execution of predefined actions through certain eye behaviors.

Except for applications handled in relatively up-to-date reviews [1-8], numerous studies have been presented in recent years. For instance, Wang et al. trained a gaze-guided attention network with X-Ray images that both reduces the dataset preparation time and enhances classification accuracy with specific configurations [9]. In addition, Sun et al. achieved high accuracy in identifying perpetrators and distinguishing innocents in eye movement analysis [10]. Issever et al. [11] measured the cognitive load of computer programmers according to their demographic and professional features when solving some object-oriented code tasks. Alternatively, Vinuela-Navarro et al. [12] investigated the effects of post-COVID-19 conditions (PCC) with saccades, fixation, and pupil responses. The participants with PCC are prone to longer latencies in some saccadic paradigms, weaker fixation stability and closer eye positions for vergence. In another application, Xu et al. [13] designed a wheelchair controlled by eye movements and based on a learning model for individuals with ALS disease [13]. Despite limitations, these applications offer promising models for the future, where eye movements could serve as auxiliary or co-control mechanisms rather than main control interfaces. Furthermore, control applications such as surgical robot control [14] and bedridden patients show the impact of tracking eye movements [15].

The pupil, a fundamental feature in the field of eye tracking, forms the cornerstone for designing studies in this domain. This process serves as the foundation for critical eye tracking functions such as gaze point producing, fixation detection, blink detection, and angular velocity calculating; it deeply affects the system performance and accuracy [16]. Also, our experience suggests that smoothing-like filters may not be beneficial for the correction of inaccurate pupil data due to effects on accurate data, especially true saccades. Therefore, the selection of a pupil detector operating within a desired or acceptable error rate is crucial [8, 17]. Moreover, system performance is another critical factor to bear in mind. Certain fields, such as medical diagnosis and real-time device control, necessitate high-speed tracking to fulfill their objectives. Conversely, others must prioritize low resource consumption on mobile applications with limited resources. In such scenarios, the choice of a pupil detector should be guided by factors like accuracy rate, computation time, and latency duration required for the task at hand. Addressing these concerns, this study proposes a novel hybrid pupil detection approach with real-time functionality. It not only simplifies the decision-making process when choosing between various pupil detectors, each with different trade-offs between real-time capability and accuracy, but also offers a solution that can mitigate this complexity.

1.1 Literature review

Researchers have explored various methods for pupil detection [18-21], classified into two categories: i) traditional methods, and ii) machine learning models.

1.1.1 Traditional Methods

Traditional methods can also be divided into three categories: amplitude-based, edge-based, and hybrid amplitude-edge methods. Amplitude-based traditional approaches encompass various techniques. For example, Morimoto et al. detected the pupil by subtracting bring and dark areas from each other via corneal reflections [22]. Navaneethan and Nandhagopal applied morphology closing operations on binarized images [23]. Glabbur et al. [24] utilized a region-coloring approach based on connected components, merging similar regions to identify the pupil. Abbasi and Khosravi [25] tracked the pupil with a genetic algorithm after simple thresholding operation. Bonteanu et al. [26] designed a binarization process based on the first negative slope of cumulative distribution function of grayscale images. Afterwards, they used Convex Hull operation to derive pupil parameters. In addition, Wan et al. [27] proposed a horizontal weighted Haar-like feature less effected by blinking, eyelashes, and eyelids. Timm and Barth [28] devised a gradient based algorithm, described and evaluated in the study by Krause and Essig [29]. In this study, the image resolution is reduced, and a coarse pupil position is attempted to be determined. Subsequently, the approximate pupil area on the original resolution is extracted, and fine detection is performed. Manuri et al. [30] improved the Starburst algorithm by applying some pre-processing operations and the ray tracking method for fine-tuning of pupil segmentation. Among these methods, Wan et al.'s [27] study stands out as one of the most successful in the amplitude-based traditional category.

In edge-based approaches, researchers have explored diverse techniques for pupil detection (PD). For instance, ElSe focuses on selecting pupil curves among edges detected by Canny detector [19]. In another research, PuRe, combines curves among edges detected by Canny detector after some morphological operations [31]. Additionally, PuRe calculates accurate metrics according to integrity of generated ellipse. PuReST is an optimization method designed that defines a ROI to be used in the next detection cycle of the PuRe [32]. Furthermore, Susitha and Subban [33] utilizes Sobel based method to remove eyelids. Indeed, edges are scored based on their connections with other edges, with the pupil selected based on the highest score. Li et al. [34] selected a possible pupil ellipse among curves found by Canny detector. This selection requires eyeball information to select suitable pupil curves for their geometrical structure according to the center point of the eyeball. Among these edge-based studies, Li et al.'s [34] work is highlighted as one of the most successful in the traditional category.

In hybrid traditional approaches, researchers have explored methods that combine amplitude-based and edge-based techniques to detect the pupil. For instance, ExCuSe selects curves found by Canny detector in most darker areas and fits a suitable ellipse [19]. Alshemmary [35] use gamma correction, smoothing operations, and binary thresholding, followed by the Hough transform to detect pupil and iris areas. Bonteanu et al. [26] converted the image to its binary version, and they fitted ellipses using Convex Hull and evaluated them according to the integrity and ellipticity of each ellipse. If the evaluation outcome is insufficient, the same image is subjected to binarization with a lower threshold value. Kassner et al. [36] found the curves with the Canny detector in the region with low amplitudes. Lastly, they defined the ellipse parameter with or without a combination of pre-ellipse parameters fitted to these curves. Notably, one of the most successful studies in the hybrid traditional method category was by Bonteanu et al. [26].

In brief, amplitude-based methods assume that the pupil is the darkest area in the image, edge-based methods select the most elliptical region as the pupil, and hybrid methods combine both amplitude and edge information. The point to note here is that the reasons of the inaccurate results produced by methods using edge information cannot be foreseen. Underlying this lies the ability of things that can move over time, such as shadows, eyelids, eyelashes, and reflections, to instantaneously form elliptical shapes. Also, considering the possibility of the eye camera moving, these things don't even need to be moving. Amplitude information, on the other hand, offers precision by eliminating areas darker than the pupil before the experiment or configuring the method based on a relevant subject to obtain accurate pupil information.

1.1.2 Machine learning models

Learning models offer a versatile approach in PD, extending beyond merely finding the pupil to encompass the detection of various eye features, including the eyeball, iris, gaze vector, eye corner points, sclera, and eyelid. While learning models can be trained to perform the specific task of PD by directly calculating the center point coordinates or segmenting the pupil region, the latter is more common. Because the size of the pupil ellipse is a crucial factor in certain subject analysis studies [37, 38]. Additionally, since a learning model may utilize both amplitude and edge information, precautions may not be taken in the initialization step of the approach to avoid inaccurate results. Indeed, there's a trade-off between traditional methods and learning models, balancing accuracy against computation time. While traditional methods run real-time with some localization lacks, learning models predict accurate with longer execution times.

One of the most popular learning models, Fuhl et al. [39] designed the Pistol model between ResNet-18 and ResNet-34 to detect pupil, iris, eyelids, and other eye features such as eye opening, gaze vector and eyeball. Chinsatit and Saitoh [40] classified images as open, near open and closed with AlexNet, then detected pupil center points with a ConvNet trained separately for each class. Lee et al. [41] suggested a fixed-sized patch with 9 cells pointing to the pupil area for remote trackers. The median cell should have the lowest amplitude average in this design to be able to contain the pupil segmented with ResNet. Chen et al. [42] trained a model named PCR-Net to detect pupil center and radius via 7 points placed between two eye corners. Gou et al. [43] developed an encoder-decoder network named Multiscale Attention Link for remote trackers to obtain pupil center points. Alternatively, Shi et al. [44] proposed the LVCF model containing V-Net for segmentation and LSTM for tracking. Wang et al. [45] predicted pupil center point and pupil radius instead of ellipse parameters, using Res-Net and Vision Transformers. Many studies also use UNet for segmentation with different hyperparameter optimizations and dataset preferences [7, 8, 46-48].

However, these models often require a large number of accurately labeled pupil images. This situation can pose challenges in terms of both training durations and labeling costs. Considering the dataset quality is not sufficient, a suitable model cannot be obtained. Although transfer learning is employed to overcome this problem, this approach may not entirely solve the relevant issue. In response to this negativity, Guo et al. [49] proposed a segmentation framework using Swin Transformers. Here, images were randomly masked with patches to achieve higher success with a smaller training set. Afterwards, segmentation was made with a network that combines Swin-Transformer blocks and UNet. Maquiling et al. [50] and Niu et al. [51] generated synthetic images based on light and reflection intensities using a Gaussian distribution, testing them with VR-based real images. They also leveraged the Segment Anything Model (SAM) for notable results with VR images, showcasing SAM's potential in eye region annotation and segmentation [52].

Some other studies applied various processing methods directly to segmentation maps of a learning model. These methods were generally like operations applied in traditional methods, presenting a rather unconventional way of enhancing model outputs using traditional techniques. As an example, study within this scope, Kim and Lee [53] selected the most elliptical area segmented by DeepLab v3+ as the pupil and used a type of interpolation for ellipse parameters with low segmentation map information using next and previous parameters. Gowroju and Kumar [48] proposed to use morphological processing to the segmentation map of UNet to perform a fine-tuning. Moreover, researchers aimed for faster performance by adopting regression-based methods instead of larger deep learning models. For example, Gou et al. [54] used shape augmented cascade regression model initialized with synthetic eye images to find pupil center point. Xiang et al. [55] suggested a classification and regression-based model to calculate pupil center point for scale mapped images addressing different resolution concerns.

An innovative approach in PD diverges from existing studies, focusing on cropping-based models. Some researchers have employed cropping to enhance the accuracy of models, while others have utilized it for faster inference times. For the accuracy of the model, PupilNet tries to select 24×24 patches containing pupil area on 16 times down-scaled images with a CNN model, and detect pupil center point with the second CNN36. Alternatively, Antonioli et al. [56] trained two UNets (just for pupils) to find a fix-sized pupil patch that is smaller 2.67× than the original image, and to segment the pupil area on the selected patch. For inference time, Vera-Olmos and Malpica [57] presented two encoder-decoder networks to select fix-sized patch that is 17× lower than the original image, and segment it. Byrne et al. [58] combined a cropping approach with pretrained UNet based on ResNet34 encoder for synthetic images, presenting Leyes method. The proposed fixed-size cropping method can be performed in two ways: the first is with respect to the image center, and the second is based on the result from the PuRe method. According to their algorithms, if the accuracy value of PuRe is sufficient, cropping is applied, and the cropped area is given to the model. Then if the model output indicates that the pupil is close to the edges, the model is re-predicted based on a crop of the same size around the relevant region. On the other hand, in the case of PuRe accuracy is not sufficient, cropping is done centrally. Additionally, their synthetic image generation approach operates based on a Gaussian distribution, creating content based on light and reflection intensity.

According to the comparisons they have shared, studies like Fuhl et al. [39] present remarkable approaches thanks to their performances.

In the literature described above, the most recent studies, published at the end of 2023 and in the first half of 2024, include [42, 43, 50, 52, 58-63]. These studies reveal that current PD methods focus on three main directions. The first direction aims to reduce resource requirements and increase system frequency by attempting to minimize input dimensions in various ways. The second direction measures performance in pupil detection by employing different models, model components, and training methods (e.g., synthetic images) developed in the field of image processing. The third direction primarily focuses on increasing the frequency in tracking methods. Additionally, the differences between the most recent studies in literature and the method proposed in this study will be addressed in the discussion section.

On the other hand, it is important to note that we have refrained from disclosing the accuracy metrics attained in studies addressed in literature. This is because some built-in datasets yield impressive accuracy even when subjected to a simple traditional method [7]. Additionally, general datasets analyzed in these studies may contain considerably erroneous annotations [64, 65]. Consequently, there's a need for a comprehensive benchmark in PD studies, possibly using a dataset designed and validated by researchers and professionals from diverse domains such as engineering and medicine.

1.2 Motivation and contributions

LEyes stands out as the pioneering study combining a learning model with a traditional method [58]. It employs PuRe as an edge-based traditional method and UNet as the learning model, although its strategy isn't solely patch-based. Generally, this combination was specially used to expedite the training of synthetic images with diverse sensor specifications, eye appearances, and environmental conditions. However, the study doesn't thoroughly investigate the impact of the traditional method on the learning model. Also, the images were processed with the central cropping or an edge-based method which is relatively difficult to predict the potential for erroneous output. Eventually, fixed-size cropping may result in the obtained image not consisting solely of the pupil, but rather containing various other objects. Therefore, fixed-size cropping may not be the most effective way to expedite the inference process.

This study focuses on developing a real-time and accurate pupil ellipse detection pipeline for wearable eye tracking systems, addressing challenges and limitations observed in existing approaches. To achieve this goal, a hybrid approach named Retro-Oriented Mind (ROM) combines a novel method in traditional category and learning model. The traditional method aims to find a dynamically determined rectangle patch potentially containing the pupil, while the learning model segments the pupil region within the patch. Thus, by reducing the input size of the learning model through a traditional method, ROM benefits from both real-time and accurate property as in a traditional method and a learning model, respectively. Also, ROM prioritizes an amplitude-based traditional method, allowing for error pre-detection and adjustment of environmental conditions. Essentially, ROM minimizes the trade-off between frequency and accuracy by combining traditional and state-of-the-art approaches. In other words, the first mechanism of ROM truncates or prunes the problem domain and purify it from unnecessary information as much as possible, and the second mechanism fine-tunes the process to achieve precise localization.

Another objective of the study is to introduce a new concept named Pupil Ellipse Trend Analysis (PETA). PETA is the way to measure detection accuracy for outputs of the learning model and to correct pupil ellipse parameters using previously parameters if detection accuracy metrics are insufficient. By using PETA as a post-processing step, it is desired to prevent especially inaccurate jumping data which may be observed because of a high relative angular velocity, deficient pupil information, unwanted shakes of the eye camera, and wrong detection maps.

As the contributions we present:

- We introduce a new concept ROM that combines traditional methods and learning models based on PD domain knowledge and causality. Also, unlike many existing studies, we don't directly utilize UNet as the learning model for ROM. Instead, we compare the performance of existing segmentation models, selecting the most suitable one.
- Although amplitude-based traditional methods have roughly less attention compared to others, they are the only category of methods that allow for definitive precautions to counter factors that may negatively affect detection accuracy. However, ensuring that the pupil is the darkest area in the images may not always be feasible. In this study, unlike other similar studies, the proposed amplitude-based traditional method in the scope of ROM does not assume the pupil to be the darkest amplitude. It combines amplitude, size, and position information to enable successful rough detection even when this darkest area criterion is not met. Thus, it also aims to enhance the effectiveness of accuracy-boosting precautions that can be taken before the session.
- Unlike previous studies with fixed input sizes, we enable encoder-decoder architectures in ROM to work with different image sizes using our proposed patching process. Uniquely qualifying the patching for real-time operation, we also arrange learning model parameters for real-time performance in accordance with the average patch size of the proposed traditional method.
- Introducing the PETA concept, we propose two new accuracy metrics and evaluate the model outputs with these metrics. Unlike past studies using various morphological methods on segmentation maps [48], we evaluate metrics directly from segmentation maps. We also design a procedure to improve inaccurate results by evaluating segmentation output with these metrics, addressing the common issue of jumping data in PD.
- We introduce an automatic configuration mechanism to be used defining traditional method settings. Thereby, we foresee that unsuitable configurations such as so high or low thresholds that can be defined by users may be prevented using this mechanism. In contrast to our work, previous auto-configuration selection techniques in PD remained at the sensor level and were not addressed at the application level [66].

The organization of this paper is outlined as follows: Section 2 elucidates the design of ROM and PETA components, while Section 4 provides an analysis of the proposed study based on the initialization and setup steps detailed in Section 3. Section 5 offers a comparison between our detector and other study, and Section 6 presents in an in-depth discussion of the findings and implications. Lastly, Section 7 addresses a conclusion of the study.

2. METHODOLOGY

The system's architecture comprises three key components: *i*) the traditional method, *ii*) the learning model within the scope of ROM, and lastly *iii*) PETA step. The traditional method process requires a grayscale input image and a configuration (thresholding etc.) to be used in finding pupil patches and exports a patch that will be in reduced resolution compared to the input image. Subsequently, learning model receives the patch and uses it to segment pupil area within the patch. Next, fitting an ellipse to segmentation map is actualized here. The PETA component comes into play by calculating accuracy metrics using the ellipse parameters and potentially making corrections if necessary. For an illustrative depiction of this architecture, given Figure 1, which represents the system as a top-level activity within an SADT (Structured Analysis and Design Technique) diagram [67].

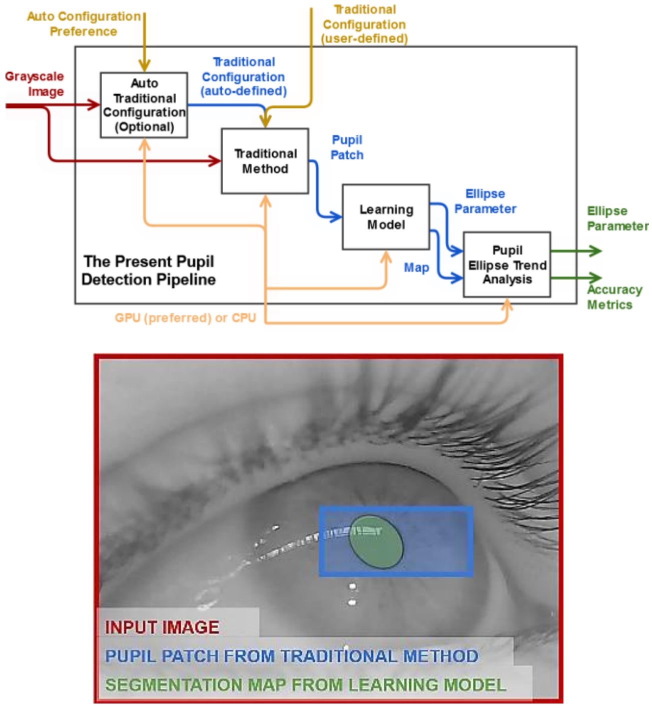

Auto Configuration Preference
Traditional Configuration (user-defined)
Grayscale Image
Auto Traditional Configuration (Optional)
Traditional Configuration (auto-defined)
Traditional Method
Pupil Patch
Learning Model
Ellipse Parameter
Ellipse Parameter
Map
Pupil Ellipse Trend Analysis
Accuracy Metrics
The Present Pupil Detection Pipeline
GPU (preferred) or CPU
INPUT IMAGE
PUPIL PATCH FROM TRADITIONAL METHOD
SEGMENTATION MAP FROM LEARNING MODEL

Figure 1. The general structure of the present study as a top-level activity of SADT diagram and the visualized result according to the pupil patch received from the traditional method and ellipse parameters calculated by the model step

It's essential to acknowledge that manually determining the traditional method configuration may lead to the patch containing insufficient or no pupil region when the image contains a pupil. This negatively impacts detection accuracy. To address this issue, an optional mechanism has been added to the system. This mechanism works at a desired time interval (e.g., every second) in a separated thread. It takes the input image which it receives at the end of the time interval and actively seeks to identify the traditional method configuration that optimally suits the given image. Thus, it tries to provide reliable and precise results.

2.1 Traditional method

The causes of inaccurate results in amplitude-based traditional methods are more predictable compared to edge-based and edge-amplitude-hybrid traditional methods. Consequently, amplitude-based traditional methods may allow the user to define configuration or initialization settings in a more controlled manner. Believing in the potential success of amplitude-based methods when used correctly, we propose an amplitude-based method for the traditional method part of the study. The proposed method has three sequential subparts:

(i) Finding Minima Pixel Sequences: The lowest amplitudes are obtained in each row for the x-axis and in each column for the y-axis. They are collected in a sequence form as given in the expression below. I_{ij} denotes pixel values of a received image. Since the sequences were created using all rows or columns of the image, each index of these sequences corresponds to a coordinate point, and the sequences include high-variance noise.

$$S_{row} = \begin{bmatrix} min\left(I_{1,1}, I_{1,2}, \cdots, I_{1j}\right) \\ min\left(I_{2,1}, I_{2,2}, \cdots, I_{2j}\right) \\ \vdots \\ min\left(I_{i,1}, I_{i,2}, \cdots, I_{ij}\right) \end{bmatrix}, \quad S_{col} = \begin{bmatrix} min\left(I_{1,1}, I_{2,1}, \cdots, I_{i1}\right) \\ min\ \left(I_{1,2}, I_{2,2}, \cdots, I_{i2}\right) \\ \vdots \\ min\ \left(I_{1,j}, I_{2,j}, \cdots, I_{ij}\right) \end{bmatrix} \tag{1}$$

(ii) Noise Reduction: To mitigate the noise present in the minima sequences, the Savitzky-Golay (SG) method was applied as a smoothing filter [68]. The filter fits a local polynomial to a window of adjacent data points and then uses the coefficients of this polynomial to compute the smoothed value at the center of the window. Filtered output of i^{th} sample $\hat{S}_i$ is calculated by a weighted summation of input signal neighbors. The coefficients C_j are determined by solving the least squares polynomial fitting problem for each window. In SG method, window length parameter (2L+1) is individually set to 20% of the length of each sequence.

$$\hat{S}_i = \sum_{j=-L}^{L} C_j \times S_{i+j} \tag{2}$$

(iii) Patch Selection: Patch selection is carried out using the smoothed sequences. The configuration settings to be taken from the user or to be defined by auto traditional method configuration mechanisms are used in this step. Three parameters, each associated with a function taking a sequence as input, govern this step: THM for thresholding, PFX (Pupil Founder X) for the x-axis sequence, and PFY (Pupil Founder Y) for the y-axis sequence.

Table 1. THM functions to be used for the selection of threshold value. A value calculated by a THM function is used for sequences of x and y pair of an input image

TH1(S)=max(S)-std(S)	TH2(S)=(MMM×0.66)+min(S)
TH3(S)=max(avg(S), MPM/2)	TH4(S) = avg (S)
TH5(S)=min(avg(S), MPM /2)	TH6(S)=(MMM×0.33)+min(S)
TH7(S)=(avg(S)+min(S))/2	TH8(S)=min(S)+std(S)
where MPM = max(S)+min(S), MMM = max(S)–min(S)	

By using a function from THM function list in Table 1, threshold values are defined dynamically for each denoised minima sequence. THM functions and equations taking a sequence shown as S are the below, roughly from highest to lower amplitude values.

Following the smoothing of the minima sequences, the process continues with the application of PFX and PFY to the x-axis and y-axis sequences, respectively. There are a total of five distinct functions, collectively referred to as Pupil Founder (PF) functions, which can be chosen for PFX or PFY. G_k expression is intended for use in relevant functions can be calculated as follows:

$$G_k = \{s_i \mid s_i < \tau, s_{i+1} < \tau, \ldots., s_{i+n} < \tau\} \tag{3}$$

where, G_k represents the grouping of consecutive sequence elements, and s_i , s_{i+1} , …, s_{i+n} are the elements of the sequence. The condition, s<τ is applied to each element in the group, ensuring that they are all smaller than the threshold value τ.

1- Median First (MF): It selects the part closest to the middle index under the threshold of the sequence on the relevant axis. It can be calculated as in Eq. (4):

$$P_{MF} = \text{argmin}_\text{k}(\text{min}\ (|G_i - M|, |G_{i+1} - M|, \ldots, |G_{i+n} - M|)) \tag{4}$$

where, M represents the median index of the S array. Accordingly, MF is useful when the objective is to position the pupil near the center of the sequence along the relevant axis. This option can be particularly useful in places where the experimental procedure is well-defined, such as laboratory environments.

2- Max Depth (MD): MD chooses the segment with the lowest amplitude value from the parts under the threshold on the relevant axis. The equation is given in Eq. (5) and it is particularly useful when the pupil is known to exhibit the darkest amplitude. This option carries the potential of obtaining a patch size smaller than expected by preventing the inclusion of other small objects besides the pupil in the patch scope in experimental or in-house environments.

$$P_{MD} = \text{argmax}_k |\, min(G_k) - \tau| \tag{5}$$

3- Max Length (ML): ML selects the segment with the highest number of elements among those below the threshold on the axis in use. This function given in Eq. (6) is suitable when the pupil is known to be larger than other dark areas.

$$P_{ML} = \text{argmax}_\text{k}(|G_k|) \tag{6}$$

4- Max of Length and Depth Product (LD): LD assigns a score to the parts under the threshold on the axis it is used and selects the part with the highest score. Score to be calculated with the expression is the product of the number of elements of the part and its absolute distance from the threshold level of the lowest amplitude. Hereby, this function given in Eq. (7) can be used when the pupil is known to be darker or larger.

Thus, it can ensure that the dark but small areas outside the pupil and the areas that are not as dark as the pupil but occupy a large area are not included in the patch.

$$P_{LD} = \text{argmax}_\text{k}(|G_k| * |\, min(G_k) - \tau|) \tag{7}$$

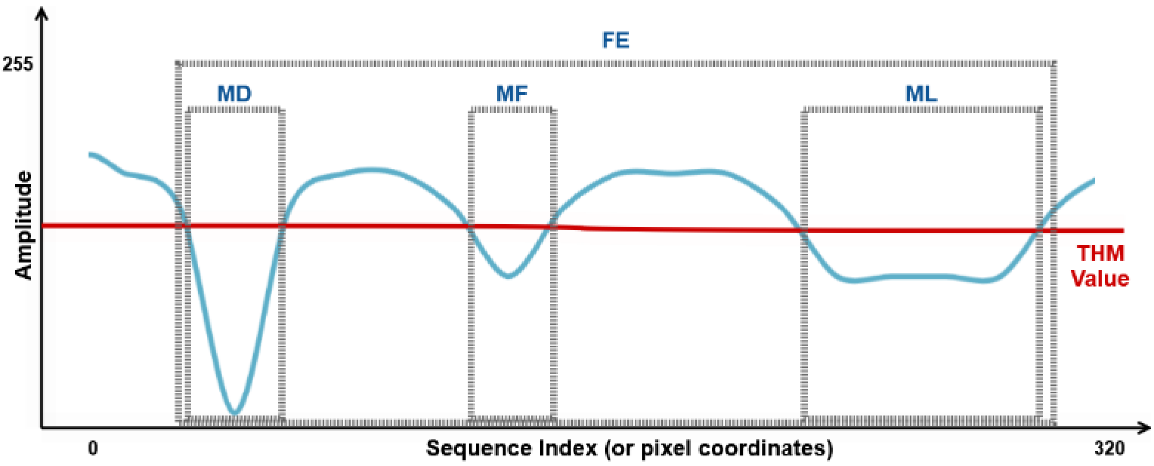


Figure 2. A denoised minima sequences for *x*-axis and different patches selected by PF functions
(Basically, these functions try to select one segment between parts under the threshold value)

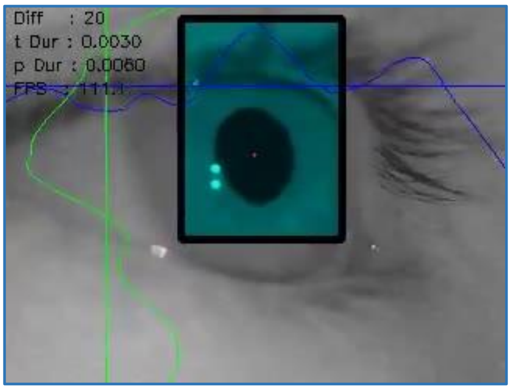


Figure 3. An example of the selected patch with randomly defined configurations

5- First and End (FE): FE selects the entire area between the first and last indices under the threshold in the axis on which it is used. The equation given in Eq. (8) is beneficial in cases where perceiving the pupil based on amplitude information alone is challenging and may result in an enlarged patch size. In general, this should be the last alternative to be chosen, indicating the need for pre-experimental preparation for other options to work effectively. However, in an environment where pupil in-formation is adequately detected, it still has the potential to produce results like the previous functions.

$$P_{FE} = \{s_n, s_{n+1}, \cdots, s_m\} \tag{8}$$

where, $n = \min\{j \mid s_j < \tau\}$ and $m = max\{j \mid s_j < \tau\}\}$.

These selections except for LD have been shown via an example minima sequence in Figure 2. Additionally, groups G, which remain below the threshold, can also be observed in the relevant example interpretation. To complement these descriptions, a visualization in accordance with the result of the traditional method was given in Figure 3. Blue and green lines show the thresholds for x and y axis, respectively. Blue and green lines indicate the minima sequences for this image. Top and right edges represent zero values for these graphs and distance from the relevant edge visualize the value of sequence indexes. The turquoise rectangle is the output of the traditional method, defined with minima sequences represented as blue and green curves. Eventually, this rectangle is sent to the learning model step.

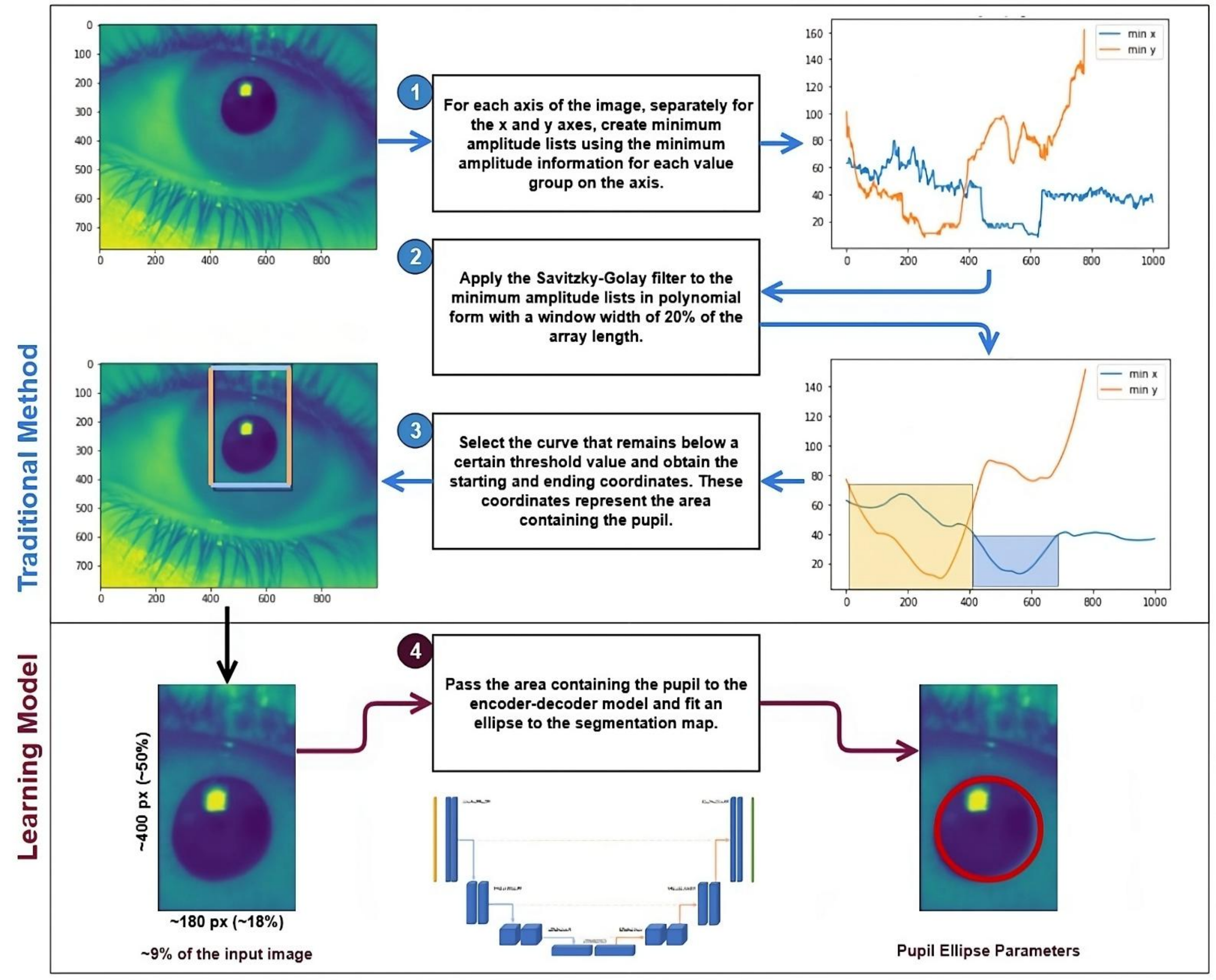
2.2 Learning model

In the learning model segment, an extensive comparison has been conducted among various segmentation models, including Unet [69], UNet++ [70], SegNet-VGG-19 [71, 72], TransUNet [73], DeepLabv3-ResNet-50 [74, 75], DeepLabv3-MobileNet-L [74, 76], PPMobi-leSeg-Tiny [77], CCSGD-ResNet-34 [78] and EGEUNet [79].

In pupil detection, UNet is a commonly used model in the literature [69], while UNet++ can create a more comprehensive map by using skip connections [70]. Similarly, TransUNet, by utilizing transformer architecture, can generate a much more comprehensive and sharp-edged map compared to Unet [73]. DeepLab, with the power of atrous and atrous spatial pyramid pooling, has the potential to produce precise results [74]. Although SegNet shows similarities to UNet, it can produce results at lower inference times thanks to its pooling indices [71]. PPMobileSeg, with its pixel-level segmentation capabilities, can yield accurate results, especially in cases with insufficient pupil information [71]. On the other hand, CCSGD operates as a very recent medical image processing approach, designed to work successfully with low parameter counts based on shallow features [78]. As for EGEUNet, it is a relatively recent model potentially applicable to detecting relatively small medical objects [79].

The performance assessment encompassed evaluating prediction time and memory consumption on GPU and CPU for both full resolution and the average resolution of the results obtained from the traditional method. In addition, the models with satisfactory dice scores have been fine-tuned and brought to a level that can operate in real-time at the average resolution of the traditional method by decreasing their parameters in various ways. In the study, fine-tuned model names are indicated by the suffix "-S".

According to the average patch size that will be addressed in the Experimental Result section, we have changed model structure as in the below: UNet-S has 3 layers with channels (32,64), (64,128), (128,256) with double convolution. Double convolutions with channel (1,32) are performed before and after from layers. UNet++-S has 3 layers with a channel size as (32,64,128,256). TransUNet-S has 128 hidden size, 128 MLP dimensions, 2 heads with channel 128 and 2 layers for the transformer architecture. Also 16 patch size, decoder channels as (64,64,64,64), skip channels as (512, 256, 64, 32), number of layers (1,1,1) for ResNet and number of skips 3 were performed. SegNet-VGG-19-S has a four VGG stages.



1 For each axis of the image, separately for the x and y axes, create minimum amplitude lists using the minimum amplitude information for each value group on the axis.

2 Apply the Savitzky-Golay filter to the minimum amplitude lists in polynomial form with a window width of 20% of the array length.

3 Select the curve that remains below a certain threshold value and obtain the starting and ending coordinates. These coordinates represent the area containing the pupil.

4 Pass the area containing the pupil to the encoder-decoder model and fit an ellipse to the segmentation map.

Figure 4. General backbone of the present pupil detector pipeline without PETA and automatic traditional configuration mechanism

Each stages have 6 VGG fea-tures. While first two stages take the features from beginning of the VGG and last stages take from the end. CCSGD-ResNet-34-S has three layers for its UNet block with (128,128,128), (128,64,128), (128,64,128) and a ResNet blocks without last 2 layers. PPMobileSeg-S has channels (8,8,8,16,32), embedding dimensions (16,32), the number of heads 2, and ½ or ¼ MobileNetV3 blocks' channel size compared with its tiny configuration.

The final step within the learning model section involves the fitting of an ellipse into the segmentation map. For this purpose, a method, operating in a least-squares manner, is employed to determine the optimal ellipse placement [80]. With reference to find an optimal ellipse, this implicit equalization is solved using a point set to be used to fit the ellipse:

$$a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6 = 0 \qquad (9)$$

Using these coefficients shown as ai, ellipse parameters can be calculated. For example, center point equalization can be given as in the following:

$$(x, y) = \left(\frac{2a_3a_4 - a_2a_5}{{a_2}^2 - 4a_1a_3}, \frac{2a_1a_5 - a_2a_4}{{a_2}^2 - 4a_1a_3}\right) \qquad (10)$$

The largest ellipse derived from the output of this method is selected and used as the pupil ellipse parameter, thereby finalizing the learning model's contribution to the overall pupil detection process. Ultimately, the calculated ellipse parameters and the segmentation map are sent to the PETA step.

Additionally, the auto traditional configuration mechanism involves running the selected learning model for the entire image and then finding the traditional method parameters that best represent the pupil parameters obtained with entire image seg-mentation. Eq. (14) in the Experimental Results section is employed for the selection of the configuration that represents the pupil area in the smallest size and pupil including rate.

With the determination of the ellipse parameters, the main structure for detecting the pupil in the study is completed. The processes from the traditional method to the learning model can generally be summarized as shown in Figure 4.

2.3 PETA

PETA approach involves two operations. The first operation computes metrics designed to assess the outputs generated by the learning model, while the second operation focuses on the development of a correction method to rectify pupil ellipse parameters, should the need arise, based on the metrics. Two metrics named entropy and intensity have been proposed for metrics of PETA. In order to calculate the entropy, a Sobel edge filter is applied to the segmentation map area only within the pupil ellipse area. To handle the limited area over the model output, map (M) is masked with ellipse parameters (E) as:

$$\widehat{M} = M \odot \mathrm{E} \tag{11}$$

After that, gradient (G) is calculated for x and y axes of $\widehat{M}$, using Sobel edge detector. Next, the sum of the gradient amplitudes $G_{i,j}$obtained at the output of the filter is divided by the total number of pixels, and entropy metric M_{Ent} is calculated as in Eq. (12).

$$M_{Ent} = \frac{1}{N}\left(\sum_{i,j} G_{i,j}\right) \tag{12}$$

In this respect, the entropy metric can indicate integrity or precision within the segmentation map from which the ellipse is generated. To calculate the intensity M_{Int} , the pixel amplitudes of the segmentation map within only the pupil ellipse area are average of $\widehat{M}$. Therefore, intensity metric can refer to the magnitude of the prediction probability in the segmentation map area from which the ellipse was generated. Using these metrics, an inference has been expressed as follows:

- *Low Entropy, Low Intensity:* The prediction is considered good. The eye may be in the half-closed position or even if the segmentation result is of low probability, it may still better compared to the alternatives.
- *Low Entropy, High Intensity:* The prediction is considered very well, and the eye is in the open position.
- *High Entropy, Low Intensity:* The prediction is deemed very poor.
- *High Entropy, High Intensity:* The prediction is considered good, and the eye is open, but it can be said that there are various obstacles or reflections on the pupil.

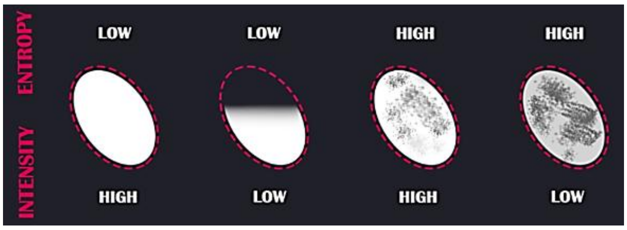


Figure 5. The visualization of Entropy and Intensity concepts (Pupil regions in a segmentation map are shown with white color. Pink ellipses represent the fitted parameters using pupil regions)

The visual representation of the above inferences is provided in Figure 5. According to this figure, when the entropy metric is low to medium and intensity is high, it indicates successful segmentation. On the other hand, the segmentation map can vary depending on the subject's eye appearance and environmental conditions. Therefore, the entropy-intensity values that will distinguish segmentation as successful and unsuccessful will also be specific to each experimental session. To dynamically determine this range and differentiate a successful segmentation, a formula is used based on the last 120 entropy and intensity values (or those within the about last 1 second, to be defined according to the system frequency) with newly calculated values. The comparison simply attempts to detect anomalies in entropy and intensity by using variance. In this way, entropy anomalies that suddenly rise too much or intensity anomalies that suddenly falls too much can be detected. Accordingly, the pupil ellipse is considered successful if $M_{Ent} > max(L_{Ent}) + var(L_{Ent})$ and $M_{Int} < min(L_{Int}) + var(L_{Int})$ and unsuccessful if it is false. L_{Ent} and L_{Int} refer to last 120 data points for entropy and intensity metrics, respectively.

In the PETA correction step, the last successful ellipse parameter is used instead of the pupil ellipse marked as unsuccessful. However, a condition has been set for this use, and any ellipse parameters that do not meet this condition are not corrected with the previous successful one. The relevant condition is that the difference between the unsuccessful ellipse center point and the last successful ellipse center point is greater than 19 pixels. The threshold value of 19 pixels is the upper band value obtained according to the preparation procedures of the LPW dataset [81]. The threshold value is grounded in the following assumptions and reasoning:

(i) Eyeball Area Within Camera Detection: It is assumed that the area of the eyeball can fit within the area that the eye camera can detect. For example, if the image collected by the eye camera is 320×240 pixels, the eyeball is expressed with roughly 240×240 pixels.

(ii) Maximum Angular Velocity of the Human Eye: A human eye can have a maximum angular velocity of 700 degrees per second [82]. Since this is a peak value allowed to only 25 degrees of visual angle, we decreased it as maximum 500 degrees for LPW dataset [83].

(iii) Frame Rate of LPW Dataset: LPW dataset is 120 Hz. Therefore, the maximum angular velocity that a human eye can achieve between two frames is 4,166 degrees (500/120).

(iv) Upper Coordinate Distance: In a 2D image, the largest difference in pixel coordinates resulting from a change in eyeball angle between two frames occurs when the geometry forms an isosceles triangle, as illustrated in Figure 6. Consequently, we computed a maximum shift of 8,723 pixels for an angular change of 4,166 degrees.

(v) Upper Band Threshold Values: Upon identifying that the maximum distance between two center points differed by approximately 9 pixels, we applied the widely accepted 5-pixel margin of error as per the literature for each center point. When we account for this 10-pixel permissible error difference (adding 5 pixels to each center point), the value utilized in our study was became 19 pixels for the upper bound. The relevant equalization rounded to nearest can be seen in the below. ω_{max} , f_{cam} , r_{eye}, $E[\varepsilon]$ parameters denote 500, 120, 120, 5 values, respectively.

$$Thr_{upp} = \left\lfloor \sqrt{2r_{eye}{}^{2}\left(1 - \cos\left(\frac{\omega_{max}}{f_{cam}}\right)\right)} + 2E[\varepsilon] + 0.5 \right\rfloor \tag{13}$$

The upper band threshold value, denoted as Thr_{upp} provide a solution to the problem of jumping data, especially during blinking or in cases where insufficient pupil information occurs due to any obstacle. However, the value of 19 pixels is not applied to pupil ellipse parameters that are marked as successfully/normal according to entropy and intensity values. In this way, if it is assumed that the segmentation is fulfilled successfully according to the metrics, the difference between the calculated and annotated pupil ellipses is prevented in cases such as the movement of the eye camera or the displacement of the pupil center point when the eye is reopened.

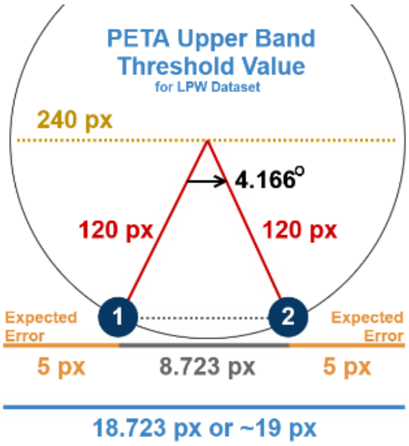


Figure 6. PETA upper bound threshold value representation between two consecutive center points for LPW dataset. The value is about 9 pixels, however we defined it as 19 pixels together with 5-pixel errors

3. EXPERIMENTAL SETUP

The development and analyses in this study were conducted using Python v3.10 and PyTorch v1.13 on a laptop with a NVIDIA RTX 3050 GPU 4GB, an AMD Ryzen 7 5800H CPU and 16 GB RAM. Two datasets, annotated by Fuhl et al. [84], were chosen for this research: the LPW dataset for training and testing purposes and the Dikablis dataset for validation. Both datasets were prepared with conventional head mounted eye trackers under the IR illumination and with real subjects in different user and environment conditions. LPW was constructed using recordings from 22 participants of various nationalities, each with different eye-region characteristics (e.g., make-up, contact lenses, glasses), captured under a wide range of everyday indoor and outdoor illumination conditions [81]. Thereby, the training dataset contains considerable real-world diversity, as it includes recordings from multiple participants, environments, usage and luminance conditions., according to the inherent variability of the LPW. While LPW was generated with 66 sessions from 22 subjects and with an eye camera running at 120Hz [81], Dikablis contains various videos, collected at 25 Hz, where the information about which user belongs to which session is unknown. Dikablis is a combined dataset consisting of data from ElSe, ExCuSe, PupilNET, and a driving study, collected from 30 participants [19, 21, 85]. These datasets were specifically designed to include challenging real-world examples, featuring low contrast, difficult lighting conditions, reflections, and cases where the pupil is not clearly visible. The data was gathered across various daily life activities, including both indoor and outdoor scenarios, driving, reading, walking, and more. Overall, the most noticeable difference between these two sets lies in the frequency of eye camera and the amount of illumination. In LPW, the eye region is generally well-illuminated, while Dikablis exhibits insufficient illumination. Additionally, both sets predominantly feature content where the pupil is visible. In other words, there are minimal obstacles between the pupil and the camera, resulting in either no or very rare occurrences of blinking during a session. Furthermore, there is often a slight pupil reflection and distinct dark areas aside from the pupil, with this condition being more prevalent in Dikablis. Therefore, Dikablis proves to be a more challenging dataset for methods relying on the amplitude information. Also, resolutions are 640×480 and 384×288 pixels for LPW and Dikablis respectively.

Because of time and physical resource concerns in the study, both dataset resolutions were reduced to 320×240 from 640×480 and 384×288 pixels. Additionally, due to the same concerns, samples that are only below of 100 MB in Dikablis set were used. However, it's worth noting that samples from participants and video pairs with identifiers 1-9, 20-12, 22-7, 22-8, 22-9, 23-3, 23-5, 24-11, 24-5, 24-8, and video pairs of the T series were excluded from the validation set since there were no corresponding annotation files available in the Dikablis dataset. Consequently, 286 videos from Dikablis were analyzed.

In the training process, videos belonging to the first 5 participants in LPW dataset were allocated for test set, while all videos from the other participants were used for training set. This approach ensured that each sample and participant were exclusively used within only one set. Each training was completed with Adam optimizer with learning rate 1e-3, ReduceRLOnPlateau with patience of 2, dice scoring as loss function and shuffled epochs with batch size of 1 [86]. Augmentation techniques such as horizontal and vertical flips, random rotation, and transpose were applied to the training set, while only horizontal flips were used for the test set. Training was monitored based on dice scores at the end of each epoch, and the process was stopped if there were no changes in dice scores for both the training and test sets over the last 6 epochs. In measurement of pupil detection accuracy, a 5-pixel error rate was performed to accept the results as successful. In the process of assessing learning models between each other, we used an LPW variation created at 3 Hz for shorter training durations, not the entire LPW dataset. In other words, the LPW-3Hz set was created by taking the first of every 40 consecutive images. Additionally, LPW variations with different fixed resolutions have been generated for training of the selected learning model. These variations included resolutions of ×2 (320×120px), ×4 (160×120px), ×6 (107×120px), ×9 (107×80px), ×16 (80×60px), ×25 (64×48px), with each set obtained by cropping the original images to encompass the entire pupil areas. Furthermore, a variation named LPW-AV containing all images from these variations has been generated. However, the model with AV was trained for only 15 epochs due to its time consumption. For visualization purposes, example samples from the LPW×4 set and LPW×16 set are displayed in Figure 7.

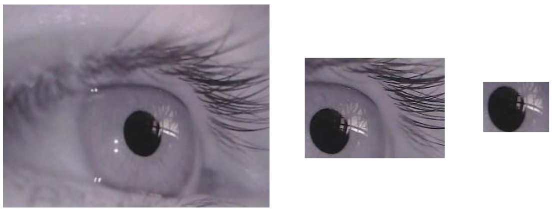


Figure 7. Example samples from LPW variation: full resolution, ×4 and ×16, respectively
Source: subject=1, video=1, image=1

As size of these datasets, 130,856 images for each LPW variation are handled without masks- total size for all variations (LPW-AV) is 915,992 images. Also 320,200 images have been used in Dikablis set without masks. Additionally, LPW-3Hz contains about 3,271 images without masks. Thereby, total size of datasets used in this study were generated with 1,239,463 images.

4. EXPERIMENTAL RESULTS

The analysis of the present traditional method for LPW dataset are given in Table 2 with details. The table presents average measurements in terms of patch size, contained pupil area (CPA) and success. Patch size is the ratio of the generated patch resolution to the original image resolution. Contained pupil area indicates the percentage of the pupil is present in the generated patch. In this analysis, we assume that the ellipse can be fitted if at least 40% of the pupil area is in the patch. Success is obtained according to this assumption. A patch was considered successful if it contained at least 40% of the pupil area and unsuccessful otherwise.

Consequently, the success value represents the ratio of successful detections to the total number of detections. To show the maximum positive effects of the traditional method, the PFX, PFY, and THM functions used were selected manually. For this, all sessions were analyzed with each function combination and the functions that would give the best performance were selected by the authors. Basically, the following equalization has been calculated for all combinations and configurations, with the highest score has been selected for each different LPW session. Also in the auto configuration mechanism, the same regime was processed to find traditional configurations for the segmentation map generated using full image received every once in a certain period.

$$Score = (100 - Patch\,Size) * CPA \tag{14}$$

On average, the patch resolution was 7.66 times smaller (13.05% of original resolution) than the original image resolution, containing 95.51% of the total pupil area, and achieving a patch generation success rate of 99.23%. However, in the next analyses, we will use the average patch size as 6×, not 7.66×, for ease of calculation. In addition, the Cumulative Distribution Function (CDF) curve of the average pupil patch percentages calculated on a video basis is given in Figure 8.

Accordingly, although the average patch percentage value is 13.05% and the average magnification factor is 7.66×, within the scope of the CDF, half of the videos have been reduced to a percentage of 7.58% or smaller. Thus, for half of the videos, the magnification factor is equal to or greater than 13.2×. On the other hand, the probability related to the intuitive 20% pupil size applied when determining the window size of the Savitzky-Golay filter is also shown in the graph. According to this, a 20% window size on each axis results in a patch that is 4% of the total image. Accordingly, in 23% of the images analyzed within the scope of LPW, a patch size of 4% or smaller, and thus a magnification factor of 25× or larger, has been achieved. The maximum and minimum magnification factors obtained within the scope of LPW are 46.9× and 2.2×, respectively. However, the accumulation in the distribution generally occurs at patch sizes of 2% and 8%, where the graph accelerates rapidly.

Table 2. The results of the traditional method according to parameters selected manually for each session of LPW

Subject	Video	PFX	PFY	THM	Patch Size (%)	CPA(%)	Success (%)	Subject	Video	PFX	PFY	THM	Patch S. (%)	CPA (%)	Success (%)
1	1	FE	FE	TH7	14.51	90.88	100.0	12	1	LD	ML	TH7	6.89	93.66	99.4
1	4	FE	MF	TH7	11.14	95.66	100.0	12	2	ML	LD	TH7	3.44	91.83	100.0
1	9	MF	MD	TH7	3.07	93.08	100.0	12	9	MF	FE	TH7	6.58	94.10	100.0
2	4	ML	ML	TH7	4.19	99.58	100.0	13	1	MF	FE	TH5	12.77	99.32	100.0
2	10	MD	ML	TH7	7.65	100.00	100.0	13	2	ML	ML	TH7	7.56	94.38	100.0
2	13	ML	ML	TH7	2.39	99.72	100.0	13	9	MD	ML	TH1	22.22	99.96	99.9
3	16	MD	MF	TH1	44.19	95.38	92.5	14	10	FE	FE	TH7	18.69	92.31	100.0
3	19	FE	ML	TH4	35.67	95.40	100.0	14	17	MF	FE	TH3	27.68	99.54	99.3
3	21	MF	FE	TH6	8.45	95.56	100.0	14	22	FE	FE	TH7	17.61	90.81	100.0
4	1	FE	ML	TH5	31.27	99.54	100.0	15	1	ML	ML	TH7	5.88	97.08	100.0
4	2	MF	ML	TH7	3.99	92.16	99.5	15	2	ML	ML	TH7	7.09	99.08	100.0
4	12	FE	ML	TH7	17.51	94.29	99.7	15	7	ML	ML	TH7	5.29	98.61	100.0
5	6	MD	FE	TH7	5.88	94.38	100.0	16	1	ML	ML	TH7	4.81	99.83	100.0
5	10	ML	MD	TH5	14.45	97.54	100.0	16	2	ML	ML	TH7	3.87	99.94	100.0
5	11	LD	MD	TH6	7.58	99.50	100.0	16	13	ML	ML	TH7	4.44	100.00	100.0
6	2	MD	ML	TH7	4.22	95.77	100.0	17	3	MF	ML	TH4	22.08	99.68	100.0
6	5	MF	ML	TH7	3.76	96.72	99.8	17	5	MF	ML	TH6	17.43	99.02	99.5
6	13	MD	MD	TH7	3.86	98.68	100.0	17	12	FE	MF	TH4	44.55	99.17	100.0
7	15	FE	FE	TH6	26.53	97.70	95.6	18	2	MF	FE	TH7	5.62	91.46	100.0
7	18	FE	MD	TH7	5.29	96.61	100.0	18	7	MF	ML	TH7	8.98	95.19	100.0
7	21	FE	MD	TH7	10.46	99.97	100.0	18	11	FE	ML	TH5	21.73	100.00	100.0
8	2	MF	ML	TH7	3.09	94.88	100.0	19	2	ML	ML	TH7	5.47	83.10	100.0
8	7	LD	ML	TH7	8.42	95.54	100.0	19	3	ML	ML	TH7	3.39	75.79	100.0
8	9	FE	FE	TH5	19.73	99.56	100.0	19	6	MF	ML	TH4	15.58	99.39	99.4
9	16	ML	ML	TH7	8.44	96.81	100.0	20	3	MF	LD	TH7	2.82	91.04	100.0
9	17	ML	ML	TH8	3.09	96.58	100.0	20	4	ML	ML	TH7	5.72	95.19	100.0
9	18	ML	ML	TH8	3.65	97.53	100.0	20	7	FE	ML	TH8	3.23	93.32	100.0
10	1	MF	ML	TH7	4.48	97.08	100.0	21	4	LD	MD	TH7	3.26	96.84	100.0
10	8	MF	ML	TH7	8.47	96.28	100.0	21	11	MD	ML	TH7	4.34	91.64	100.0
10	11	FE	FE	TH3	46.00	99.76	99.9	21	12	FE	ML	TH8	2.13	83.12	100.0
11	2	FE	FE	TH7	12.16	92.98	100.0	22	1	MF	MF	TH1	34.80	92.34	91.3
11	7	MF	ML	TH7	10.34	92.44	100.0	22	2	MF	MF	TH1	43.98	94.58	91.3
11	13	MF	ML	TH4	30.66	99.06	100.0	22	17	MF	FE	TH2	36.84	86.16	82.6

The examination of the selected learning models within the scope of the study is presented in Table 3. Generally, the models produced similar dice scores for the LPW-3Hz dataset. The source of this similarity is that the LPW dataset contains pupil information that can be well detected by a learning model. On the other hand, small differences in scores distinguish the models from each other. This difference arises from cases that are relatively less common in the LPW dataset. Examples of such cases include blinking, strong reflections, and a large angle of the gaze vector relative to the eye camera. Therefore, small improvements in scores indicate that the model is more robust in inadequate pupil information. Hence, UNet++, UNet++-S, TransUNet, and TransUNet-S models are the best-performing models in terms of both dice and test dice scores. Therefore, it can be said that any of these four models can be used for pupil detection.

Another goal of this study is to aim for a real-time (at 120 Hz and above) and less resource-consuming system. GPU performances in terms of inference time and memory requirements for both original resolution (320×240px) and the resolution of the average patch size (107×120px). According to the measurements, UNet-S, UNet++-S, SegNet-VGG-S and CCSGD-RN-34-S perform shortest inference time, while UNet-S, SegNet-VGG-S, TransUNet-S, all PPMobileSeg models, and EGEUnet require less resource area.

The learning model part of the study will be run in GPU. However, investigating models on only CPU may put an insight off related to sufficiency of models for devices with less capacity such as mobile phones. CPU analysis includes similar measurements like the GPU analysis. The relevant performance metrics are shown in Table 3. Accordingly, CCSGD-RN-34, CCSGD-RN-34-S, EGEUNet and all PPMobileSeg models perform fairly successfully results while compared to others.

In the stage of the learning model phase of the ROM concept, the UNet++-S model was selected after reviewing the relevant analyses. This selection was made because the UNet++-S model has produced successful Dice scores and stands out in terms of resource requirements. This model can operate in approximately 6 milliseconds on a GPU using pupil patches and requires 340 MB of space, achieving 95.6% success in training and 86% in testing. Consequently, the UNet++-S model has been used in all other analyses as the learning model. However, it can be said that the CCSGD-RN-34-S model is more prominent in CPU usage due to the balance between speed and performance.

Table 4 provides training sessions of the UNet++-S model with different LPW variations and their corresponding test measurements. According to this, there is a significant correlation between the resolution in the training set and the test dice score of UNet++-S. In general, as the difference between the resolution in the training set and the test input resolution increased, the dice score decreased. On the other hand, LPW-AV has been able to produce successful results in all variations since it was trained with all training variations. Additionally, training UNet++-S with all variations (LPW-AV) has also increased the test dice score of each variation except x6. The inability to significantly increase the x6 test dice score may be related to the fact that the training with LPW-AV was limited to 15 epochs. However, this situation was not considered, and the UNet++-S model trained with LPW-AV was used in the continuation of the study.

Table 3. Inference time, consumed memory, and dice metrics for selected learning models trained with 3Hz LPW

Model and Device		Inference Time (ms)		Memory (GB)		Param (M)	LPW-3Hz		
		Full	×6	Full	×6		Dice	Loss	Test Dice
UNet [87]	GPU	28.42	7.13	1.71	0.38	17.2	0.95	0.032	0.842
	CPU	521.38	91.25	1.58	0.26				
UNet-S	GPU	11.05	4.25	0.93	0.17	1.9	0.954	0.029	0.843
	CPU	204.69	27.94	0.89	0.14				
UNet++ [88]	GPU	39.77	10.35	2.64	0.51	9.1	0.958	0.025	0.868
	CPU	681	102.53	2.3	0.37				
UNet++-S	GPU	25.41	5.75	1.88	0.34	2.2	0.956	0.027	0.86
	CPU	391.63	56.48	1.68	0.24				
SegNet-VGG-S	GPU	15.06	3.16	1.13	0.2	0.3	0.933	0.047	0.763
	CPU	234.65	28.71	1.13	0.19				
Trans UNet [89]	GPU	42.46	26.24	2.07	0.46	105.1	0.954	0.028	0.876
	CPU	948.19	225.96	1.8	0.37				
Trans UNet-S	GPU	14.74	8.09	0.88	0.19	3.2	0.952	0.03	0.867
	CPU	199.14	38.72	0.82	0.16				
DeepLabv3-RN-50 [90]	GPU	25.34	13.8	1.85	0.38	39.6	0.943	0.053	0.838
	CPU	440.27	96.36	1.54	0.27				
DeepLabv3-MNv3-L [90]	GPU	11.72	11.07	0.39	0.07	11	0.902	0.104	0.78
	CPU	64.39	20.36	0.33	0.06				
PPMobile Seg-Tiny [91]	GPU	20.36	19.72	0.25	0.04	0.6	0.899	0.106	0.778
	CPU	43.34	17.69	0.27	0.04				
PPMobile Seg-S [91]	GPU	19.98	19.66	0.1	0.01	0.1	0.889	0.111	0.764
	CPU	21.22	15.51	0.1	0.01				
CCSGD-RN-34 [92]	GPU	11.73	9.8	0.8	0.2	22	0.943	0.035	0.872
	CPU	102.57	30.88	0.6	0.1				
CCSGD-RN-34-S	GPU	5.76	5.24	0.4	0.08	1.4	0.946	0.033	0.868
	CPU	47.47	11.88	0.3	0.06				
EGE UNet [93]	GPU	15.12	15.27	0.1	0.01	0.05	0.93	0.042	0.808
	CPU	31.96	17.85	0.1	0.02				

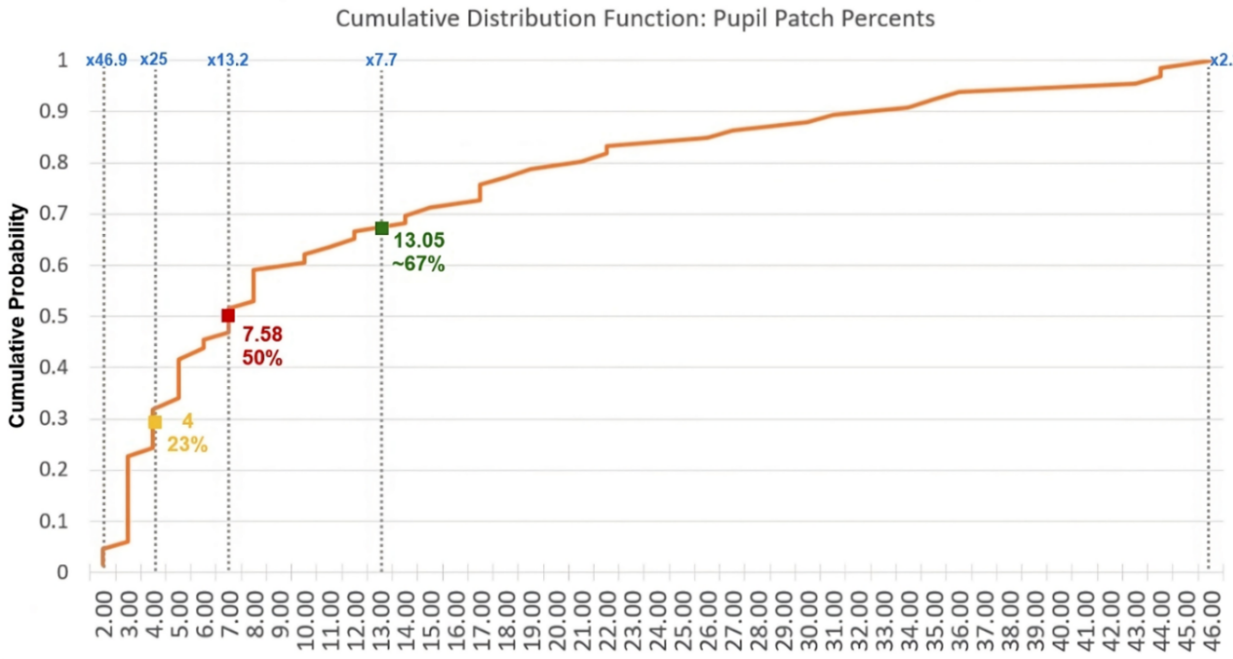


Figure 8. Cumulative distribution function for average pupil patch percent of the traditional method

Table 4. UNet++-S model training with different variations of the LPW dataset to display the effect of the input size to the accuracy

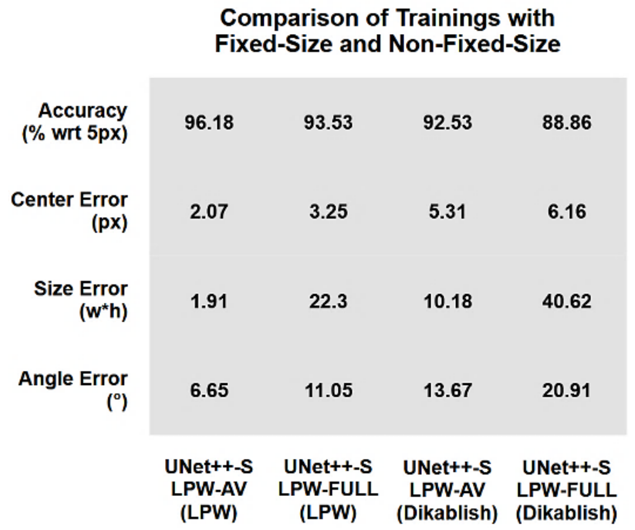
				UNet++-S				
Dataset Variation	**LPW-Full**	**LPW-2×**	**LPW-4×**	**LPW-6×**	**LPW-9×**	**LPW-16×**	**LPW-25×**	**LPW-AV**
Dice	0.962	0.96	0.961	0.961	0.96	0.961	0.96	0.963
Loss	0.019	0.02	0.019	0.019	0.019	0.019	0.019	0.018
Test Dice-×1	0.892	0.855	0.798	0.776	0.684	0.553	0.427	0.896
Test Dice-×2	0.878	0.894	0.834	0.883	0.799	0.726	0.643	0.901
Test Dice-×4	0.775	0.81	0.903	0.785	0.865	0.833	0.728	0.906
Test Dice-×6	0.778	0.873	0.816	0.909	0.817	0.816	0.739	0.899
Test Dice-×9	0.625	0.75	0.863	0.836	0.896	0.889	0.827	0.908
Test Dice-×16	0.353	0.6	0.799	0.832	0.881	0.898	0.854	0.909
Test Dice-×25	0.077	0.416	0.597	0.775	0.767	0.854	0.893	0.909
Avg Test Dice	0.625	0.742	0.801	0.828	0.815	0.795	0.730	0.904

In the next analysis, the impact of the study components on accuracy was assessed, and relevant measurements are presented in Table 5. While L and D refer to LPW and Dikablis datasets, LM and TM refer to the learning model and traditional method, respectively. AUTO denotes the automatic traditional method configuration mechanism. In addition, measurements were made for two alternatives of AUTO that run every 100ms and 333ms. The results were calculated based on a 5-pixel error and average ellipse parameter error. Ellipse error consists of pixel errors in center point (x, y), size (major, minor) and angle error, respectively. During the analysis, it was seen that the traditional method's patch area may not be at a sufficient level for the learning model and ellipse fitting operations. Therefore, the patch taken from the traditional model was analyzed by expanding it to 20px from the bottom and left, and 10px from the top and right edges for LPW. If an edge does not have enough space to be expanded, the corresponding edge is expanded to the maximum possible space. As a result of LPW analyses, the extended traditional model patch has a resolution of 19.50% compared to the original image, corresponding to an average resolution of ×5 for the learning model. For Dikablis, average patch size was 10.27% and CPA amount was 94.60%, by using traditional configurations in accordance with scores of each combination. In measurements made with LPW for the average patch size to be similar, the patch area was expanded by 30 pixels from the bottom and 22 pixels from the left in the vertical direction. Therefore, the results were approximately taken as having ×5 resolution for Dikablis as well. Furthermore, since the Dikablis dataset's preparation frequency of 25 Hz, the upper band value was set at 52 pixels for the Dikablis, according to the calculation explained in Eq. (13).

Also, in the analysis of the LPW and Dikablis datasets, three procedures were applied. Firstly, no binarization operation was applied to the segmentation maps when the results were obtained. In other words, the maps obtained from the model were used directly for the analysis. Secondly, the model trained with LPW-AV was used in all analyses, including ones without traditional method. Normally, one might consider using the model trained with LPW-Full for the analyses without the traditional method because LPW-AV was generated for models predicting in accordance with different input resolutions. However, as shown in Figure 9, we discovered that LPW-AV has a data augmentation effect. Training the model with different resolutions, even though it operates at a fixed resolution, results in a significant improvement in accuracy. In the figure, four performance metrics are shown. Accuracy represents successful detections based on a 5-px error value. Center error is the average error between the estimated and annotated ellipse center coordinates. Size error is presented as the product of the average errors in width and height between the estimated and annotated ellipses. Angle error represents the average angular errors between the estimated and annotated ellipses. Lastly, if the width or height values of the ellipse parameters calculated in Dikablis are smaller than 5 pixels, the previously calculated ellipse was used. With the use of the 5-px size regime, the UNet++-S trained with LPW-AV achieved an accuracy rate of 92.53%, while in the case where it was not used, the accuracy rate was 92.47%. Although the success difference is quite low, a size-dependent adjustment in models trained with different resolutions, such as LPW-AV, may have the potential to increase success to a certain extent.

Table 5. Accuracy metrics for the combination of the study components

LM	TM	PETA	AUTO	L-5px (%)	D-5px (%)	L-Ellipse Error (Average)	D-Ellipse Error (Average)
✓	✗	✗	✗	96.18	92.53	(1.35, 1.30), (1.25, 1.53), 6.65	(4.64, 2.60), (2.97, 3.43), 13.67
✓	✗	✓	✗	96.23	92.56	(1.32, 1.28), (1.24, 1.52), 6.63	(4.60, 2.58), (2.96, 3.42), 13.66
✓	✓	✗	✗	95.11	92.60	(2.18, 2.03), (1.64, 2.16), 8.03	(4.64, 2.96), (2.52, 3.41), 13.44
✓	✓	✗	**100 ms**	93.41	92.19	(3.29, 2.61), (1.86, 2.64), 8.76	(5.91, 3.27), (2.60, 3.44), 13.87
✓	✓	✗	**333 ms**	91.67	91.34	(4.64, 3.53), (2.29, 3.33), 9.82	(6.54, 3.71), (2.74, 3.64), 14.42
✓	✓	✓	✗	95.15	92.71	(2.10, 1.99), (1.63, 2.15), 8.02	(4.51, 2.89), (2.50, 3.38), 13.40
✓	✓	✓	**100 ms**	93.66	92.30	(3.10, 2.48), (1.81, 2.56), 8.61	(5.76, 3.19), (2.57, 3.41), 13.81
✓	✓	✓	**333 ms**	92.01	91.38	(4.44, 3.39), (2.19, 3.20), 9.56	(6.43, 3.70), (2.73, 3.62), 14.39



Comparison of Trainings with Fixed-Size and Non-Fixed-Size

	UNet++-S LPW-AV (LPW)	UNet++-S LPW-FULL (LPW)	UNet++-S LPW-AV (Dikablish)	UNet++-S LPW-FULL (Dikablish)
Accuracy (% wrt 5px)	96.18	93.53	92.53	88.86
Center Error (px)	2.07	3.25	5.31	6.16
Size Error (w*h)	1.91	22.3	10.18	40.62
Angle Error (°)	6.65	11.05	13.67	20.91

Figure 9. Comparison of models trained with LPW-Full and LPW-AV. Accordingly, LPW-AV, containing different resolution of same images

Table 6. Average GPU pred. times of UNet++-S for input resolutions at different subsampling rates in millisec

×1(Full)	×2	×4	×5	×6	×9	×16	×25
25.41	13.25	7.51	6.12	5.75	5.62	5.09	4.90

Consequently, according to Table 5, over 92% was achieved for both LPW and Dikablis datasets. In a situation where traditional model configurations were determined manually, only a 1% 5-px loss of accuracy was observed compared to a flow where only the learning model was used, for LPW dataset. Furthermore, it is observed that the automatic configuration mechanism has relatively little impact on accuracy. The patch size obtained is the same as the patch rate obtained from AUTO, but the success rate obtained from AUTO may be 2.5% lower. Especially, the fact that success does not significantly decrease with the use of this mechanism was crucial for the usability of this mechanism. Thereby this situation may reduce the need for the user to adjust on the traditional method.

Table 6 presents the prediction time of the UNet++-S model at different resolutions. Specifically, the selected model exhibits runtimes of 25.41 milliseconds at full resolution, 5.57 milliseconds with the average patch size (×6), and 6.12 milliseconds with the extended average patch size (×5).

Table 7. Average GPU execution times of the study sub-components except for the learning model in millisec

Code	Sub-Component	Time (ms)	Code	Sub-Component	Time (ms)
S1	Finding minima sequences	0.10	**S6**	Fitting ellipse to a map with x5 resolution	0.04
S2	Smoothing with Savitzky-Golay filter	0.72	**S7**	Selecting traditional configuration automatically	8.88
S3	Selecting patch with PFX, PFY, THM functions	0.12	**S8**	Converting matrix to tensor for images with x5	0.04
S4	Calculating entropy & intensity	0.25	**S9**	Converting tensor to matrix for images with x5	0.01
S5	Detecting edges with Sobel filter for 10% area	0.13			

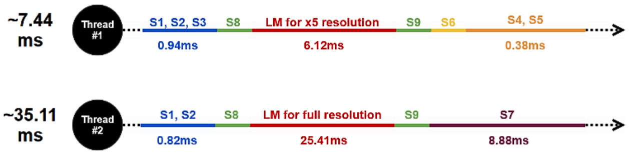


Figure 10. Execution times of the pipeline working with two threads for detection and auto configuration processes

Table 8. Worst case in performance changes of auto-configuration mechanism and its comparison with ROM and ROM-PETA model

Model	Time (ms)	FPS	Accuracy (%)
ROM (R)	6.5	153	92.60
ROM – PETA (RP)	7.5	133	92.71
AUTO (R) – 10 Hz	7.4	135	92.19
AUTO (RP) – 10 Hz	8.4	119	92.30
AUTO (R) – 3 Hz	6.7	149	91.34
AUTO (RP) – 3 Hz	7.7	130	91.38

Table 7 displays the GPU execution times for various subcomponents used in the study. Apart from these, the running time of some mathematical operations and control expressions that will require less computing power has been ignored. Figure 10 provides a graphical representation of the execution times for the proposed pipeline on the GPU. Notably, the pupil detection process requires approximately 7.5 milliseconds, while the auto-configuration mechanism, which operates on a separate thread, requires 35 milliseconds. Consequently, it can be inferred that pupil detection can be performed at a rate exceeding 120 Hz.

Table 8 presents the worst-case runtimes of the auto-configuration mechanism under two variants: ROM-only (R) and ROM combined with PETA (RP). As noted in Table 7, the automatic selection of traditional method parameters typically completes within 8.88 ms. Since this selection process occurs on the CPU while segmentation is executed on the GPU, parallelism ensures that the auto-configuration mechanism introduces minimal latency to the overall detection process. However, when auto-configuration is not executed during the segmentation but only after segmentation and by processing only a few parts of the selection in each frame, its frequency affects inference time and FPS. For example, an AUTO-10Hz setting implies one selection operation every 100 ms, whereas AUTO-3Hz corresponds to a 333 ms interval. Under this scheduling, auto-configuration introduces an average latency of approximately 0.9 ms per frame at 10 Hz and about 0.1 ms at 3 Hz. Moreover, the less frequently traditional parameters are recalculated, the more the performance tends to degrade. As shown in the table for the Dikablis dataset, reducing the auto-configuration frequency from 10 Hz to 3 Hz results in roughly a 1% drop in accuracy. Nonetheless, using the 10 Hz variant leads to only a modest decrease in core performance, approximately 0.40% - which may be considered an acceptable trade-off for maintaining high responsiveness.

5. COMPARISON

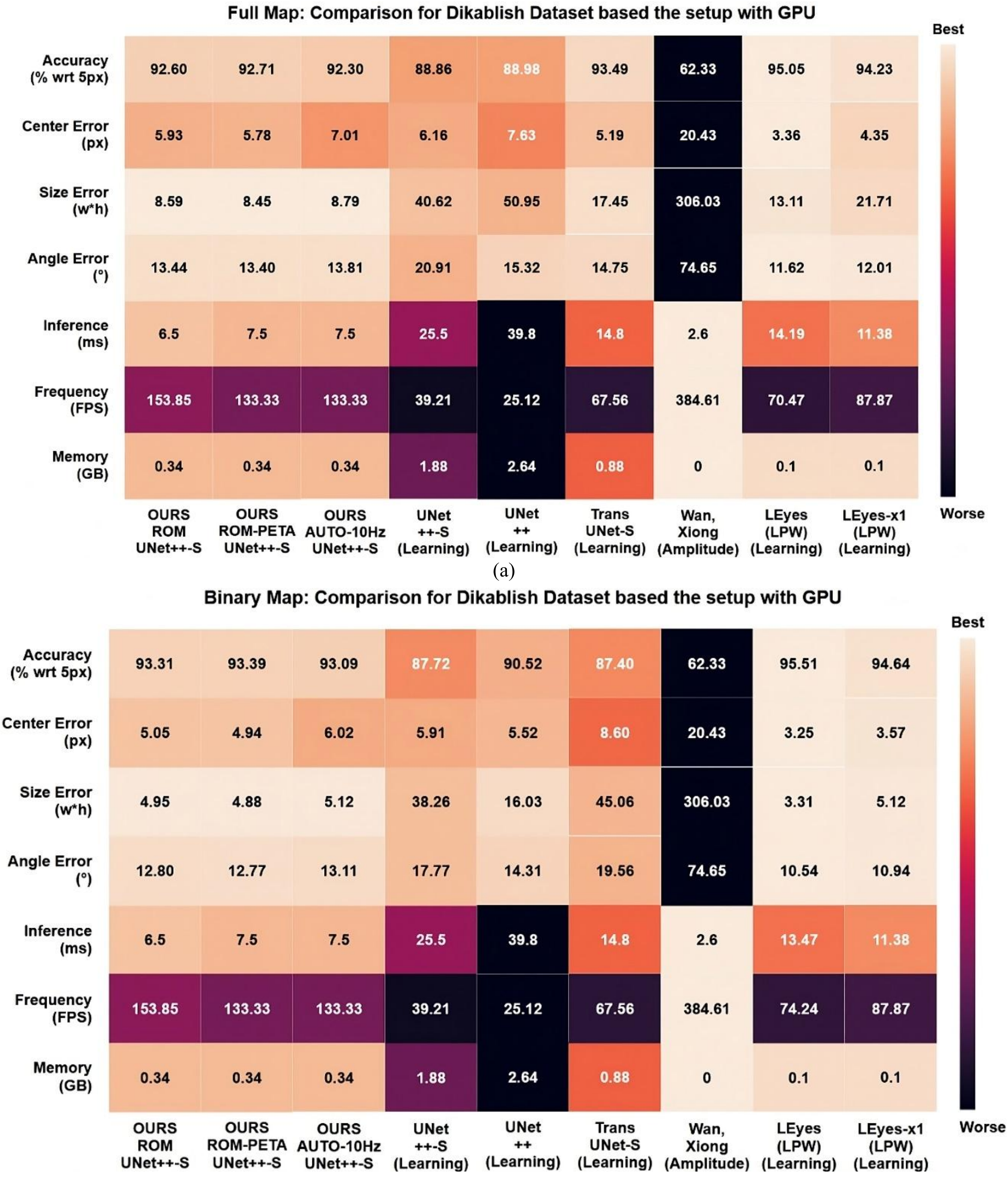
In this section, the performance of the proposed traditional method and learning model in the ROM-PETA concept is compared with previous studies. For this purpose, the comparison incorporates learning models such as LeNet-5, along with findings from recent studies [87, 94, 95]. Other potentially successful edge-based and amplitude-edge based methods could not be included as their code has not been shared before in a public repository. Also, Pistol has not been analyzed since, along with the pupil, it can make many other detections such as iris and sclera. Additionally, the analyses include segmentation models trained with LPW-3Hz and seen to be successful in Experiment section. These models were trained according to the standards specified in the Experimental Setup section with LPW-Full.

In the comparison, LEyes' model initially trained with EDS2019 synthetic images yielded unexpectedly negative results, with 69.20% accuracy for a 5px error. To further explore the potential success of Leyes in pupil detection on real images, the model was retrained on LPW. Our Experiment section findings indicate that an encoder-decoder model trained with fixed resolutions may not accurately detect inputs of different resolutions. Therefore, training LEyes with LPW-Full (320×240), as desired to work at a resolution of 128×128, may not accurately reflect the model's performance. Following these considerations, a new variation of LPW has been created with a resolution of 128×128, with the pupil position located at the center of the cropped area. Except for resolutions, the regime specified in the Experimental Setup section was applied during training. The PuRe and central cropping approach, as applied in LEyes' evaluation, were followed as mentioned in the study published by Byrne at al. [58]. It's noteworthy that another contribution of LEyes is its ability to generate successful results from a small number of synthetic images created based on parameters determined considering environmental conditions and imaging sensor characteristics during an experimental session. The unexpected result with the EDS2019 model might be due to differences in these parameters. Therefore, due to the differences in dynamics between LPW and Dikablis, we didn’t use LEyes' synthetic image generation approach when retraining and directly trained the model with the new LPW variation.

The comparison results were presented in Figure 11. There are two different analyses: full map and binary map. In full map, segmentation map received the relevant model was used to analyses directly. In binary map, values less than 126 in the segmentation map have been set to 0, while values greater than or equal to 126 have been set to 255. Additionally, there are two different versions of LEyes. In the original LEyes analysis, as mentioned in the literature, the model can make predictions twice for a single image. Suggested by us, LEyes-×1 produces single prediction, and if the calculated ellipse center is too close to the edges of the patch, it does not make a second prediction. Models prefixed with "ours" were all been trained with LPW-AV, while models other than LEyes were trained with LPW-Full. Since the potential for success improvement with LPW-AV was presented in Figure 9, this complicates the comparison between "ours" models and those trained with LPW-Full. However, LPW-AV is currently the only alternative for making predictions with variable patches and is a necessity for our proposed models. Therefore, LPW-AV was not used in models not designed to work with variable resolutions. Also, in the Dikablis dataset with 320,200 images, while the LEyes model made 107,907 repetitions in the full map, it made 73,539 repetitions in the binary map. The LEyes model works alone in 8.17 milliseconds, while PuRe works in 3.2 milliseconds. Therefore, the inference time of LEyes was calculated considering both the prediction times of PuRe and the model, as well as the number of repetitions it made on average.

The results reveal that the choice between using a full or binary map is model-dependent. Although the full map is more efficient in some models, especially in our study and LEyes, using the binary map is a better choice when making predictions. Additionally, both our study and the LEyes models work much more efficiently compared to a directly used segmentation model. This indicates that a learning model with a traditional method can operate more successfully with less unnecessary information. Additionally, it can be argued that it may not be necessary for the LEyes model to make repetitions in predictions.

Thus, with a slightly shorter inference time, an increase of approximately 17 FPS has been achieved. However, for better results in the repetition-free version of LEyes, using the binary map might be more favorable. In the comparison between our study and LEyes, it is observed that LEyes works slightly more accurately. This level of accuracy holds true for center error, size error, and angle error for the repetitive version of LEyes.



(a)

(b)

Figure 11. Comparison of our study and selected studies based on full or binary segmentation maps

On the other hand, our proposed study operates at almost twice the speed. Especially in slightly more controlled experimental environments, it can be said that the small success difference between our study and LEyes will completely close. However, our approximately two times faster prediction speed will remain constant. In the case where PETA metrics and the correction approach are added to our study, predictions can still be made at over 120 FPS, while performance metrics have improved only slightly. Additionally, our automatic configuration mechanism operates at almost similar levels of success. Thus, it has the potential to eliminate the need for users to make manual settings. When examining the only traditional method in our comparison, as expected, it operates very fast but is much less successful compared to learning models. In general, learning models, on the contrary, are more accurate but slower. This situation more clearly demonstrates the trade-off between learning models and traditional methods.

6. DISCUSSION

Although their advantages, amplitude-based methods may fail due to pupil-like dark regions or reflections from objects between the camera and eye. Success in edge-based methods depends on maintaining the pupil's elliptical form; ghosting and varying eye-closing can lead to detection errors. Hybrid methods can fail if amplitude and shape information don't complement each other. While ROM uses amplitude-based techniques, it reduces errors from darker non-pupil regions through user-defined functions as seen in Figures 12(a)-(c). However, issues from objects like glasses reflections remain (e.g. Figure 12(e)). Moreover, since the pupil is searched within a restricted area, ROM can be more robust than edge-based methods during blinks when pupil shape information decreases, reducing the occurrence of jumping data. On the other hand, this capability may lead to fewer errors in images where amplitude or ellipticity is dominant (e.g., Figures 12(b)-(c)), such as in hybrid models. This is because the pupil is not solely searched in the darkest region, and when searching for the pupil within a dark area of the image, the process relies not only on shape information but also on the combined consistency of both features.

Pupil shape information can degrade under conditions like blinking and ghosting (secondary reflections), negatively impacting edge-based methods. While deep learning models also rely on shape cues, ROM's traditional method can be more robust in such cases due to its constrained pupil search region. As shown in Figure 12(c), Figure 12 (f), Figure 12(g), Figure 12(h), even with significant shape loss, the detected pupil remains close to the plausible pupil area. Since the next detection is found in the selected ROI, it highlights ROM's advantage in mitigating the jumping data issue, especially when combined with the proposed amplitude-based error prevention strategy, outperforming edge-reliant approaches.

The definition of the Savitzky-Golay filter as a parabola is due to the characteristic embedding of the pupil area, which is typically oval or rectangular in the curve. However, the window size has been intuitively determined by the study authors and represents the average size of the pupil in the eye region. According to this intuition, for a ratio of 20% on each axis, the pupil area corresponds to approximately 4% of the image. In the experiments conducted, it was observed that lower determined ratios could not correct the noise, while higher ratios reduced the pupil information in the curve. Nevertheless, to achieve a better definition, it may be considered to calculate the average pupil area information for the Savitzky-Golay window size using dataset annotations.

The average patch size of the traditional method proposed within the scope of ROM is 13.05%. However, according to the CDF curve (in Figure 8), the average patch percentage for half of the LPW is equal to or less than 7.58% (approximately ×13.2). This suggests that the traditional method can produce much smaller patches for more controlled experimental procedures. As a result, the learning model can operate approximately 1ms faster (based on an ×16 magnification factor) and the system frequency can be increased to 184 FPS. Furthermore, by ensuring that the learning model deals with significantly less extraneous information, it can be used solely for localization processes.

Literature includes several datasets for pupil detection, such as CASIA, IITD, MMU, and UUTD [7]. However, MMU (995 images) and UUTD (1120 images) datasets are small and do not reflect natural eye-tracking sessions. They contain well-illuminated images with limited variation in eye openness, restricting model evaluation. CASIA was collected in a controlled and remote iris imaging setup, making it unsuitable for wearable systems. Therefore, LPW, with 130,856 images from 66 uncontrolled session videos, offers a more challenging training environment.

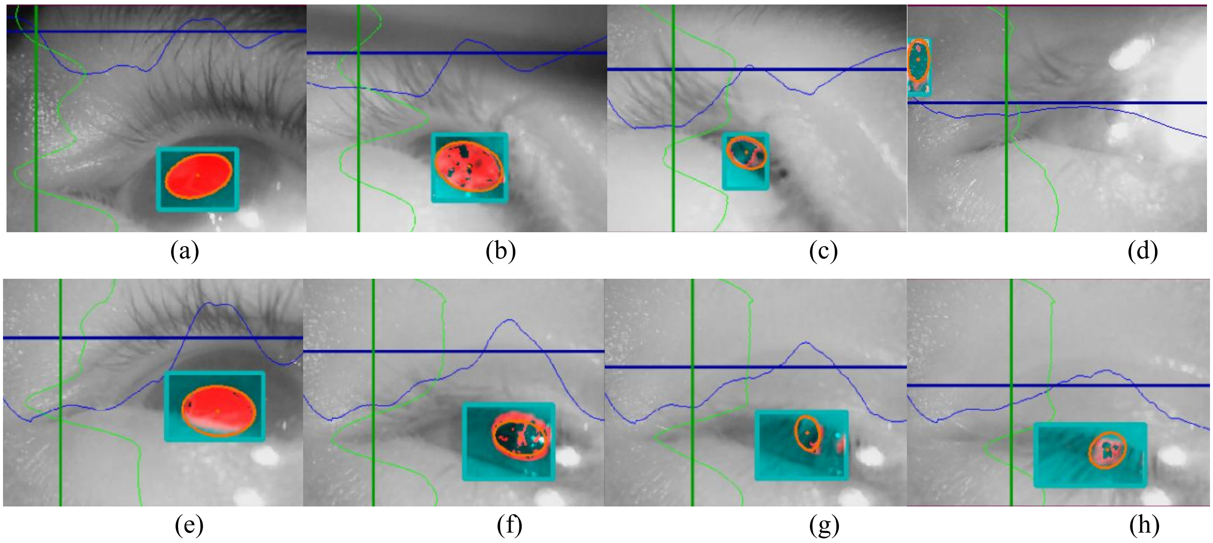


Figure 12. Example images where ROM fails and detects successfully

Dikablis (a subset of TEyeD) is larger (320, 200 images), collected from a different eye tracker, and more challenging regarding illumination. Hence, training on LPW was validated with Dikablis. The recent and popular OpenEDS dataset was developed for eye tracking in VR systems. Although suitable for evaluating the proposed method's performance in VR [84], OpenEDS was not included since our amplitude-based approach would require additional procedures to avoid pupil searches in dark regions caused by VR headsets. Based on this, we haven't analyzed VR images in the current scope of the study [96, 97].

When determining dataset variations for LPW, image resolutions were not set uniformly but were instead prepared in accordance with the CDF curve and the intuitively defined 4% pupil size information. Accordingly, variations start from the lowest magnification factor of ×2 and end at the intuitive value related to the average pupil size, which is ×25, as given in the CDF graph. Intermediate resolution variations are not created at equal intervals but are instead based on regions of accumulation in the CDF. Therefore, variations with resolutions of ×25, ×16, ×9, and ×6 are prepared for areas with high accumulation. Those with resolutions of ×4, ×2, and the original are referenced for areas with less accumulation. However, for a more comprehensive solution, resolution variations can be diversified, and more frequent resolution choices can be made even in areas with less accumulation.

Unlike edge-based methods and learning models, amplitude-based traditional methods in the literature can enable the implementation of pre-experimental precautions to enhance computational accuracy. These precautions are generally associated with the removal of regions darker than the pupil from the eye area. However, this may not always be possible, and there may be dark areas in the eye region that cannot be removed. Therefore, in such cases, it should not be expected that traditional methods using amplitude information will work accurately. The amplitude-based traditional method proposed in the ROM concept of this study does not operate on the assumption that the pupil is the darkest region. Instead, it allows the experimental operator to determine the relationship between the pupil and other dark areas and operates according to PF function selections. PF functions provide a pre-experimental precaution alternative that is more advantageous than other amplitude-based traditional methods by evaluating the darkness, length, and position information of the pupil simultaneously. On the other hand, the First and End function, which ensures the selection of all dark areas, can be used to maximize the likelihood of the pupil being included within the patch, even in extreme situations where pre-experimental precautions cannot be taken. In this regard, the proposed amplitude-based traditional method can inspire measures regarding pupil selection criteria in other methods that operate with traditional processes.

While training and comparing the learning models in the study, certain regulations were implemented to reduce the training duration on the fixed hyperparameters and the data set. For a more accurate analysis, training processes can be designed with the all the data set (not with 3 Hz set), cross-validation and hyperparameter optimizations. However, due to our temporal and physical resources, the relevant regulation had to be implemented. Also, the proposed model was tested at 320×240 resolution, commonly used in eye tracking. Evaluating its performance at different resolutions is important for robustness, and we consider this a direction for future work. On the other hand, training the proposed UNet++-S entirely on an unseen dataset (non-IR and structurally different from pupil data) is considered future work, as it would further strengthen the model's claims of generalizability.

In the literature, pupil detection successes are often reported using metrics such as center error and accuracy based on a 5-pixel error. These metrics may be inadequate for comparing models, as the mean value is insufficient to define a stochastic process producing ellipse parameters. Therefore, the comparison section includes not only performance metrics based on ellipse center points but also size and angle errors. Sharing these metrics in future studies might better reflect the performance that studies can offer. On the other hand, sharing size and angle errors is crucial for understanding the success of a segmentation model working with pupil patches. This is because, in cases where the patches created are small-sized and well-centered by the pupil, the success of the segmentation model may not be apparent. To better understand the performance of the segmentation model, average size error and the average size of the created patches can serve as good indicators.

With the operation of the ROM concept, a learning model with a traditional method can operate more efficiently with less unnecessary information. This is especially prominent in the field of remote eye tracking, as direct pupil information cannot be obtained by capturing the entire scene. Many objects in the scene may be black or circular, like the pupil. Therefore, in remote eye tracking, facial detection is first performed, and the eye region is then extracted, and lastly pupil is segmented. In other words, in remote eye trackers, several procedures are applied to ensure that learning models receive only the most relevant information. This study demonstrates that the "most relevant visual information" principle, which remote eye tracking systems already apply, can also be additionally implemented for wearable eye tracking systems.

In deep learning studies, training sets may be given to training after being cropped. However, in this type of cropping, the resolution of the cropped image is rescaled towards the fixed resolution in the dataset. Although a similar cropping approach has been applied in LPW-AV, resolutions have not been expanded to the original resolution. Additionally, LPW-AV, which includes cropped images at different resolutions, has a data augmentation effect and has been able to significantly improve performance. On the other hand, within the scope of this study, it has not been investigated whether this increase in success is due to different resolutions or only cropping. Furthermore, while exploring the impact of these two cropping approaches on pupil detection could be a different research topic, almost none of the studies in the literature have fixed-size cropping been used as a data augmentation method.

Models designed to work with pupil patches were trained with LPW-AV, which combines all LPW variations. However, this increases the training duration cost. For example, while one epoch UNet with LPW-Full is completed in approximately 3-4 hours, LPW-AV takes about 10-12 hours for one epoch. To reduce this cost during training, two regimes can be applied. The first is to select a certain number of samples from each LPW variation rather than using all variations, creating a new mixed LPW variation. The second is to complete this process with synthetic image generators. The working principle of a synthetic image generator could be to produce images at a lower resolution, where the model makes less successful predictions in real-time. We do not explore these two alternatives within the concept of reducing training times and attribute this research as future work. On the other hand, in the ROM structure, if a model is used whose success is not affected by the input resolution, there is no issue with a variation like LPW-AV and, consequently, no problem of increased training time.

The existing literature has highlighted the utilization of various eye features for diverse applications, emphasizing the potential value of exploring new features. However, our model is designed to find pupil ellipse parameters only. Making the developed model detectable for other eye features can be considered as another future work of this study.

Parameters marked as unsuccessfully according to PETA's metrics are not corrected if they do not meet the condition in the correction approach. It may be considered to improve the current correction approach to correct the unsuccessful parameters mentioned, but our study aimed to provide a solution for serious jumping data only. In addition, PETA's upper band recommendation is only for the human eye. For the developed method to be used in an animal other than a human, the assumptions will need to be updated according to the relevant animal.

The threshold of 19 pixels, as determined in PETA's upper band correction approach, is notably high, and its impact on improving performance is minute. For a more optimal upper band, the eyeball can be detected, and then the angular change between two consecutive frames can be observed. However, since our method does not detect eyeballs in its current form, it had to offer a procedure that produces higher upper band values. The integration of an eyeball detection feature or method has the potential to further enhance the accuracy of the PETA correction. However, on the other hand, this correction method with or without eyeball parameters may be useful for traditional pupil detectors. When the model is also capable of detecting the eyeball, steps 1 and 4 in Section 2.3 — which include the empirical procedures for calculating the PETA threshold — will no longer be necessary. Consequently, it becomes possible to directly apply a function like the one in Equation 13 to the desired eye tracker without relying on prior assumptions. However, integrating an eyeball detection mechanism and adapting the function to determine the upper-bound threshold based on the detected eyeball region are considered as the next steps of this study.

ROM runs at an average of 6.5 ms per frame on GPU—approximately 4× faster than UNet and UNet++-S, and 6× faster than UNet++. This corresponds to 153 FPS for ROM, compared to 35, 39, and 25 FPS for UNet, UNet++-S, and UNet++, respectively. On CPU, ROM achieves around 17 FPS, while the others drop to 1.5–2.5 FPS. While we expect similar speed advantages on such systems, measuring inference time on low-resource hardware is planned as future work to better assess ROM's scalability. ROM also requires significantly less RAM (~340 MB vs. ~1700 MB for UNet) and is compatible with compression techniques like quantization and pruning. Further acceleration is possible with TensorRT or ONNX, especially on lower-end devices.

ROM represents an approach rather than a rigid structure, allowing flexibility in the models and methods employed. According to the naming used in this study, works that combine traditional methods with learning models are referred to as ROM studies. Consequently, ROM presents an opportunity for future research, enabling researchers in the pupil detection field to explore innovative studies based on the fusion of traditional and state-of-the-art techniques. Additionally, from this perspective, we attribute LEyes' pupil detector as a study within the ROM concept.

As mentioned in the literature review, the most recent pupil detection studies are [42, 43, 50, 52, 58-63]. In our comparison with PONet, the proposed model not only performs more accurately due to its lack of limitation to 15 positions but also generates pupil area information thanks to its segmentation feature [61]. This is a crucial feature for cognitive studies. While EV-Eye provides a high-level frequency, it is more expensive, non-wearable, and likely less accurate. Therefore, pupil detection frame by frame with CMOS cameras remains a necessity. Res-CNN achieves a significant portion of its frequency increase by reducing resolution [63]. However, as resolution decreases, the error rate increases. Our proposed ROM structure, on the other hand, crops the image instead of reducing the resolution. Although the QAT structure in Res-CNN is not used in our study, it can be adapted. This would provide an additional frequency increase. Furthermore, the SAM-based model, Vir-NET, and Mal-NET can be directly expressed with the ROM structure [43, 50, 52]. Thus, it is possible to develop new studies that improve the detection speeds of all three models. A similar observation can be made for Byrne et al. [58]. For more accurate results, incorporating noise reduction and segmentation map correction operations into ROM, along with improving learning model accuracy, can be a function to be tested in future studies. However, Jamaludin et al.'s studies work according to black amplitude and lack the accuracy of a learning model. As mentioned in the literature, black amplitude information can easily produce erroneous results due to its insufficiency. On the other hand, the ROM structure is not valid for Pistol [39]. This is because Pistol focuses on detecting multiple components that can spread across the entire image, requiring the model input to be evaluated as a whole. However, as stated in the study, if a single segmentation model is used to improve performance, it is possible to add the Pistol model to the learning model part of ROM solely for pupil and iris detection. PCR-Net is also unsuitable for ROM for a similar reason as Pistol, as it focuses on detecting non-pupil eye components for remote systems [42].

7. CONCLUSION

This research introduces a pupil ellipse detector designed with the innovative concepts of Retro-Oriented Mind (ROM) and PETA. Both ROM and PETA are novel and performance improving approaches in pupil detection field. Benefitting from the proposed methodology, the detector can run real-time and accurately.

In addition to running at more than 150 Hz in the development environment of the study, the proposed detector was able to show significantly higher accuracy than directly used segmentation models. This situation is an indicator that the learning model can operate more efficiently with the less unnecessary information it receives. Therefore, the detector has the potential to be directly included in general-purpose uses of eye tracking systems. Also, the use of the study in fields that require high accuracy, such as medicine, requires the regularization of some environmental factors and experimental procedures related to eye appearance. It is possible to say that the developed detector with a correct experimental procedure has the accuracy to be used for medical purposes.

Additionally, the proposed automatic configuration mechanism provides almost similar success compared to when the most successful configuration is manually selected. Thus, the need for manually determining settings such as thresholds can be eliminated.

Furthermore, to the best of our knowledge, this study stands as one of the most high-performing detectors, achieving a harmonious balance between accuracy, latency, and frequency. Compared to LEyes, which is another work that can be evaluated in the ROM concept, our pupil detector in the ROM concept has almost similar accuracies but works about two times faster. This situation arises from both our patch approach not being fixed-sized and the model parameters being downsized to operate in real-time. It has also been observed that the ROM study we proposed works much more efficiently than using an encoder-decoder model directly.

ACKNOWLEDGMENTS

The author, Serif Inanir, expresses sincere gratitude to Yılmaz Kemal Yüce, Wolfgang Fuhl, and Sean Anthony Byrne for their objective insights and kind support during the development of this study.

REFERENCES

[1] Adhanom, I.B., MacNeilage, P., Folmer, E. (2023). Eye tracking in virtual reality: A broad review of applications and challenges. Virtual Reality, 27(2): 1481-1505. https://doi.org/10.1007/s10055-022-00738-z

[2] Brunyé, T.T., Drew, T., Weaver, D.L., Elmore, J.G. (2019). A review of eye tracking for understanding and improving diagnostic interpretation. Cognitive Research: Principles and Implications, 4: 1-16. https://doi.org/10.1186/s41235-019-0159-2

[3] Efetürk, Ö.A., Turgut, G., Dereshgi, H.A., Yılmaz, A. (2022). A review of visual attention research using eye-tracking technologies. Journal of Smart Systems Research, 3(2): 128-135.

[4] Alemdag, E., Cagiltay, K. (2018). A systematic review of eye tracking research on multimedia learning. Computers & Education, 125: 413-428. https://doi.org/10.1016/j.compedu.2018.06.023

[5] Klaib, A.F., Alsrehin, N.O., Melhem, W.Y., Bashtawi, H.O., Magableh, A.A. (2021). Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies. Expert Systems with Applications, 166: 114037. https://doi.org/10.1016/j.eswa.2020.114037

[6] Pauszek, J.R. (2023). An introduction to eye tracking in human factors healthcare research and medical device testing. Human Factors in Healthcare, 3: 100031. https://doi.org/10.1016/j.hfh.2022.100031

[7] Rathnayake, R., Madhushan, N., Jeeva, A., Darshani, D., Subasinghe, A., et al. (2023). Current trends in human pupil localization: A review. IEEE Access, 11: 115836-115853. https://doi.org/10.1109/ACCESS.2023.3325293

[8] Wang, S.Y., Ji, Y.K., Bai, W., Ji, Y., Li, J.J., Yao, Y.J., Zhang, Z., Jiang, Q., Li, K. (2023). Advances in artificial intelligence models and algorithms in the field of optometry. Frontiers in Cell and Developmental Biology, 11: 1170068. https://doi.org/10.3389/fcell.2023.1170068

[9] Wang, S., Ouyang, X., Liu, T., Wang, Q., Shen, D. (2022). Follow my eye: Using gaze to supervise computer-aided diagnosis. IEEE Transactions on Medical Imaging, 41(7): 1688-1698. https://doi.org/10.1109/TMI.2022.3146973

[10] Sun, C., Ding, N., Zhuang, D., Liu, X. (2023). Eye movement evidence in investigative identification based on experiments. Journal of Safety Science and Resilience, 4(3): 316-328. https://doi.org/10.1016/j.jnlssr.2023.07.003

[11] Issever, D., Catalbas, M.C., Duran, F. (2023). Examining factors influencing cognitive load of computer programmers. Brain Sciences, 13(8): 1132. https://doi.org/10.3390/brainsci13081132

[12] Vinuela-Navarro, V., Goset, J., Aldaba, M., Mestre, C., Rovira-Gay, C., et al. (2023). Eye movements in patients with post-COVID condition. Biomedical Optics Express, 14(8): 3936-3949. https://doi.org/10.1364/BOE.489037

[13] Xu, J., Huang, Z., Liu, L., Li, X., Wei, K. (2023). Eye-Gaze controlled wheelchair based on deep learning. Sensors, 23(13): 6239. https://doi.org/10.3390/s23136239

[14] Wang, W., Luo, Y., Wang, J., Wang, X., Song, H. (2023). Robotic-assisted laparoscopic adjustment: A meta-analysis and review. IEEE Transactions on Instrumentation and Measurement, 72: 1-19. https://doi.org/10.1109/TIM.2023.3298396

[15] Takeuchi, K., Yamazaki, Y., Yoshifuji, K. (2020). Avatar work: Telework for disabled people unable to go outside by using avatar robots. In Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, Cambridge, United Kingdom, pp. 53-60. https://doi.org/10.1145/3371382.3380737

[16] Holmqvist, K., Örbom, S.L., Hooge, I.T., Niehorster, D.C., Alexander, R.G., et al. (2023). Eye tracking: Empirical foundations for a minimal reporting guideline. Behavior Research Methods, 55(1): 364-416. https://doi.org/10.3758/s13428-021-01762-8

[17] Binaee, K., Sinnott, C., Capurro, K.J., MacNeilage, P., Lescroart, M.D. (2021). Pupil tracking under direct sunlight. In ACM Symposium on Eye Tracking Research and Applications, Germany, pp. 1-4. https://doi.org/10.1145/3450341.3458490

[18] Min-Allah, N., Jan, F., Alrashed, S. (2021). Pupil detection schemes in human eye: A review. Multimedia Systems, 27(4): 753-777. https://doi.org/10.1007/s00530-021-00806-5

[19] Fuhl, W., Tonsen, M., Bulling, A., Kasneci, E. (2016). Pupil detection for head-mounted eye tracking in the wild: An evaluation of the state of the art. Machine Vision and Applications, 27: 1275-1288. https://doi.org/10.1007/s00138-016-0776-4

[20] Păsărică, A., Bozomitu, R.G., Cehan, V., Lupu, R.G., Rotariu, C. (2015). Pupil detection algorithms for eye tracking applications. In 2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME), Brasov, Romania, pp. 161-164. https://doi.org/10.1109/SIITME.2015.7342317

[21] Swathi, A., Kumar, S. (2021). Review on pupil segmentation using CNN-region of interest. In Intelligent Communication and Automation Systems, CRC Press, pp. 157-168. https://doi.org/10.1201/9781003104599-12

[22] Morimoto, C.H., Koons, D., Amir, A., Flickner, M. (2000). Pupil detection and tracking using multiple light

sources. Image and Vision Computing, 18(4): 331-335. https://doi.org/10.1016/S0262-8856(99)00053-0

[23] Navaneethan, S., Nandhagopal, N. (2021). RE-PUPIL: Resource efficient pupil detection system using the technique of average black pixel density. Sādhanā, 46(3): 114. https://doi.org/10.1007/s12046-021-01644-x

[24] Gabbur, P., Hua, H., Barnard, K. (2010). A fast connected components labeling algorithm and its application to real-time pupil detection. Machine Vision and Applications, 21: 779-787. https://doi.org/10.1007/s00138-009-0183-1

[25] Abbasi, M., Khosravi, M.R. (2020). A robust and accurate particle filter-based pupil detection method for big datasets of eye video. Journal of Grid Computing, 18(2): 305-325. https://doi.org/10.1007/s10723-019-09502-1

[26] Bonteanu, P., Cracan, A., Bozomitu, R.G., Bonteanu, G. (2019). A robust pupil detection algorithm based on a new adaptive thresholding procedure. In 2019 E-Health and Bioengineering Conference (EHB), Iasi, Romania, pp. 1-4. https://doi.org/10.1109/EHB47216.2019.8970070

[27] Wan, Z.H., Xiong, C.H., Chen, W.B., Zhang, H.Y. (2021). Robust and accurate pupil detection for head-mounted eye tracking. Computers & Electrical Engineering, 93: 107193. https://doi.org/10.1016/j.compeleceng.2021.107193

[28] Timm, F., Barth, E. (2011). Accurate eye centre localisation by means of gradients. Visapp, 11: 125-130. https://doi.org/10.5220/0003326101250130

[29] Krause, A.F., Essig, K. (2019). Boosting speed-and accuracy of gradient based dark pupil tracking using vectorization and differential evolution. In Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, New York, United States, pp. 1-5. https://doi.org/10.1145/3314111.3319849

[30] Manuri, F., Sanna, A., Petrucci, C.P. (2020). PDIF: Pupil detection after isolation and fitting. IEEE Access, 8: 30826-30837. https://doi.org/10.1109/ACCESS.2020.2973005

[31] Santini, T., Fuhl, W., Kasneci, E. (2018). PuRe: Robust pupil detection for real-time pervasive eye tracking. Computer Vision and Image Understanding, 170: 40-50. https://doi.org/10.1016/j.cviu.2018.02.002

[32] Santini, T., Fuhl, W., Kasneci, E. (2018). PuReST: Robust pupil tracking for real-time pervasive eye tracking. In Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, Poland, pp. 1-5. https://doi.org/10.1145/3204493.3204578

[33] Susitha, N., Subban, R. (2019). Reliable pupil detection and iris segmentation algorithm based on SPS. Cognitive Systems Research, 57: 78-84. https://doi.org/10.1016/j.cogsys.2018.09.029

[34] Li, J., Li, S., Chen, T., Liu, Y. (2018). A geometry-appearance-based pupil detection method for near-infrared head-mounted cameras. IEEE Access, 6: 23242-23252. https://doi.org/10.1109/ACCESS.2018.2828400

[35] AlShemmary, E. (2020). Towards accurate pupil detection based on morphology and Hough transform. Baghdad Science Journal, 17(2): 583-583. https://doi.org/10.21123/bsj.2020.17.2.0583

[36] Kassner, M., Patera, W., Bulling, A. (2014). Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Washington, pp. 1151-1160. https://doi.org/10.1145/2638728.2641695

[37] Skaramagkas, V., Giannakakis, G., Ktistakis, E., Manousos, D., Karatzanis, I., Tachos, N.S., Tsiknakis, M. (2021). Review of eye tracking metrics involved in emotional and cognitive processes. IEEE Reviews in Biomedical Engineering, 16: 260-277. https://doi.org/10.1109/RBME.2021.3066072

[38] de Rooij, A., Wijers, I., Marinussen, M. (2021). Emergence of metacognitive knowledge via audible pupil size. In Proceedings of the 32nd European Conference on Cognitive Ergonomics, Italy, pp. 1-7. https://doi.org/10.1145/3452853.3452870

[39] Fuhl, W., Weber, D., Eivazi, S. (2022). Pistol: Pupil invisible supportive tool to extract pupil, iris, eye opening, eye movements, pupil and iris gaze vector, and 2d as well as 3d gaze. arXiv preprint arXiv:2201.06799. https://doi.org/10.48550/arXiv.2201.06799

[40] Chinsatit, W., Saitoh, T. (2017). CNN-based pupil center detection for wearable gaze estimation system. Applied Computational Intelligence and Soft Computing, 2017(1): 871956. https://doi.org/10.1155/2017/8718956

[41] Lee, Y.W., Kim, K.W., Hoang, T.M., Arsalan, M., Park, K.R. (2019). Deep residual CNN-based ocular recognition based on rough pupil detection in the images by NIR camera sensor. Sensors, 19(4): 842. https://doi.org/10.3390/s19040842

[42] Chen, J.C., Yu, P.Q., Yao, C.Y., Zhao, L.P., Qiao, Y.Y. (2024). Eye detection and coarse localization of pupil for video-based eye tracking systems. Expert Systems with Applications, 236: 121316. https://doi.org/10.1016/j.eswa.2023.121316

[43] Gou, C., Zhong, R., Yu, Y. (2023). MAL-Net: Multiscale attention link network for accurate eye center detection. Computer Vision and Image Understanding, 234: 103750. https://doi.org/10.1016/j.cviu.2023.103750

[44] Shi, L., Wang, C., Tian, F., Jia, H. (2021). An integrated neural network model for pupil detection and tracking. Soft Computing, 25(15): 10117-10127. https://doi.org/10.1007/s00500-021-05984-y

[45] Wang, L., Wang, C., Zhang, Y. (2022). Pupil detection using hybrid vision transformer. International Journal of Pattern Recognition and Artificial Intelligence, 36(12): 2255016. https://doi.org/10.1142/S0218001422550163

[46] Lee, Y., Lee, S., Jang, S., Wang, H.J., Seo, Y.J., Yang, S. (2022). Pupil detection and segmentation for diagnosis of Nystagmus with U-Net. In 2022 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Korea, pp. 1-2. https://doi.org/10.1109/ICEIC54506.2022.9748687

[47] Han, S.Y., Kwon, H.J., Kim, Y., Cho, N.I. (2020). Noise-robust pupil center detection through CNN-based segmentation with shape-prior loss. IEEE Access, 8: 64739-64749. https://doi.org/10.1109/ACCESS.2020.2985095

[48] Gowroju, S., Kumar, S. (2021). Robust pupil segmentation using UNET and morphological image processing. In 2021 international mobile, intelligent, and ubiquitous computing conference (MIUCC), Cairo, Egypt, pp. 105-109. https://doi.org/10.1109/MIUCC52538.2021.9447658

[49] Guo, Y., Yan, X., Ma, L., Tang, L., Xiong, J., Yan, Y.

(2022). A pupil segmentation framework with masked image modeling enhanced swin-transformer. In 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Las Vegas, USA, pp. 2439-2444. https://doi.org/10.1109/BIBM55620.2022.9995137

[50] Maquiling, V., Byrne, S.A., Nyström, M., Kasneci, E., Niehorster, D.C. (2023). V-IR-Net: A novel neural network for pupil and corneal reflection detection trained on simulated light distributions. In Proceedings of the 25th International Conference on Mobile Human-Computer Interaction, Athens, Greece, pp. 1-7. https://doi.org/10.1145/3565066.3608690

[51] Niu, L., Gu, Z., Ye, J., Dong, Q. (2021). Real-time localization and matching of corneal reflections for eye gaze estimation via a lightweight network. In Proceedings of the Ninth International Symposium of Chinese CHI, Hong Kong, pp. 33-40. https://doi.org/10.1145/3490355.3490359

[52] Maquiling, V., Byrne, S.A., Niehorster, D.C., Nyström, M., Kasneci, E. (2024). Zero-shot segmentation of eye features using the segment anything model (SAM). Proceedings of the ACM on Computer Graphics and Interactive Techniques, 7(2): 1-16. https://doi.org/10.1145/3654704

[53] Kim, T., Lee, E.C. (2020). Experimental verification of objective visual fatigue measurement based on accurate pupil detection of infrared eye image and multi-feature analysis. Sensors, 20(17): 4814. https://doi.org/10.3390/s20174814

[54] Gou, C., Zhang, H., Wang, K., Wang, F.Y., Ji, Q. (2019). Cascade learning from adversarial synthetic images for accurate pupil detection. Pattern Recognition, 88: 584-594. https://doi.org/10.1016/j.patcog.2018.12.014

[55] Xiang, Z., Zhao, X., Fang, A. (2022). Pupil center detection inspired by multi-task auxiliary learning characteristic. Multimedia Tools and Applications, 81(28): 40067-40088. https://doi.org/10.1007/s11042-022-12278-4

[56] Antonioli, L., Pella, A., Ricotti, R., Rossi, M., Fiore, M. R., et al. (2021). Convolutional neural networks cascade for automatic pupil and iris detection in ocular proton therapy. Sensors, 21(13): 4400. https://doi.org/10.3390/s21134400

[57] Vera-Olmos, F.J., Malpica, N. (2017). Deconvolutional neural network for pupil detection in real-world environments. In Biomedical Applications Based on Natural and Artificial Computing: International Work-Conference on the Interplay Between Natural and Artificial Computation, Corunna, Spain, pp. 223-231. https://doi.org/10.1007/978-3-319-59773-7_23

[58] Byrne, S.A., Maquiling, V., Nyström, M., Kasneci, E., Niehorster, D.C. (2025). LEyes: A lightweight framework for deep learning-based eye tracking using synthetic eye images. Behavior Research Methods, 57(5): 129. https://doi.org/10.3758/s13428-025-02645-y

[59] Jamaludin, S., Ayob, A.F.M., Akhbar, M.F.A., Ali, A.A.I.M., Imran, M.M.H., Norzeli, S.M., Mohamed, S.B. (2023). Efficient, accurate and fast pupil segmentation for pupillary boundary in iris recognition. Advances in Engineering Software, 175: 103352. https://doi.org/10.1016/j.advengsoft.2022.103352

[60] Liu, J., Chi, J., Yang, Z. (2024). A review on personal calibration issues for video-oculographic-based gaze tracking. Frontiers in Psychology, 15: 1309047. https://doi.org/10.3389/fpsyg.2024.1309047

[61] Xiong, J., Zhang, Z., Wang, C., Cen, J., Wang, Q., Nie, J. (2024). Pupil localization algorithm based on lightweight convolutional neural network. The Visual Computer, 40(11): 8055-8071. https://doi.org/10.1007/s00371-023-02946-0

[62] Zhao, G., Yang, Y., Liu, J., Chen, N., Shen, Y., Wen, H., Lan, G. (2023). Ev-eye: Rethinking high-frequency eye tracking through the lenses of event cameras. Advances in Neural Information Processing Systems, 36: 62169-62182.

[63] Ates, G.C., Coskunpinar, C., Tse, D., Pelaez, D., Celik, E. (2024). Robust residual convolutional neural network based pupil tracking for low-computational power applications. Engineering Applications of Artificial Intelligence, 133: 108235. https://doi.org/10.1016/j.engappai.2024.108235

[64] Akinlar, C., Kucukkartal, H.K., Topal, C. (2022). Accurate CNN-based pupil segmentation with an ellipse fit error regularization term. Expert Systems with Applications, 188: 116004. https://doi.org/10.1016/j.eswa.2021.116004

[65] Wibirama, S., Ardiyanto, I. (2019). An improved pupil detection method under eyeglass occlusions. In 2019 Asia Pacific Conference on Research in Industrial and Systems Engineering (APCoRISE), Depok, Indonesia, pp. 1-6. https://doi.org/10.1109/APCoRISE46197.2019.9318871

[66] Fiala, G., Ye, Z., Steger, C. (2022). Framework for image sensor design parameter optimization for pupil detection. In 2022 8th International Conference on Systems and Informatics (ICSAI), Kunming, China, pp. 1-6. https://doi.org/10.1109/ICSAI57119.2022.10005532

[67] Ahmed, F., Robinson, S., Tako, A.A. (2014). Using the structred analysis and design technique (SADT) in simulation conceptual modeling. In Proceedings of the Winter Simulation Conference 2014, Savannah, GA, USA, pp. 1038-1049. https://doi.org/10.1109/WSC.2014.7019963

[68] Raju, M. H., Friedman, L., Bouman, T.M., Komogortsev, O.V. (2023). Filtering eye-tracking data from an eyelink 1000: Comparing heuristic, savitzky-golay, iir and fir digital filters. Journal of Eye Movement Research, 14(3): 10-16910. https://doi.org/10.16910/jemr.14.3.6

[69] Ronneberger, O., Fischer, P., Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, pp. 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

[70] Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Granada, Spain, pp. 3-11. https://doi.org/10.1007/978-3-030-00889-5_1

[71] Badrinarayanan, V., Handa, A., Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv preprint arXiv:1505.07293. https://doi.org/10.48550/arXiv.1505.07293

[72] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

https://doi.org/10.48550/arXiv.1409.1556

[73] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.Y. (2021). Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306. https://doi.org/10.48550/arXiv.2102.04306

[74] Chen, L.C., Papandreou, G., Schroff, F., Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587. https://doi.org/10.48550/arXiv.1706.05587

[75] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[76] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

[77] Tang, S., Sun, T., Peng, J., Chen, G., Hao, Y., Lin, M.H., Xiao, Z.H., You, J.B., Liu, Y. (2023). Pp-mobileseg: Explore the fast and accurate semantic segmentation model on mobile devices. arXiv preprint arXiv:2304.05152. https://doi.org/10.48550/arXiv.2304.05152

[78] Hu, S., Liao, Z., Xia, Y. (2023). Devil is in channels: Contrastive single domain generalization for medical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, Switzerland, pp. 14-23. https://doi.org/10.1007/978-3-031-43901-8_2

[79] Ruan, J., Xie, M., Gao, J., Liu, T., Fu, Y. (2023). EGE-UNet: An efficient group enhanced UNet for skin lesion segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, Switzerland, pp. 481-490. https://doi.org/10.1007/978-3-031-43901-8_46

[80] Structural Analysis and Shape Descriptors. OpenCV, Ellipse Fitting in a Least-Squares Sense. https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html#gaf259efaad93098103d6c27b9e4900ffa, accessed on Oct. 10, 2024.

[81] Tonsen, M., Zhang, X., Sugano, Y., Bulling, A. (2016). Labelled pupils in the wild: A dataset for studying pupil detection in unconstrained environments. In Proceedings of the ninth biennial ACM Symposium on Eye Tracking Research & Applications, New York, United States, pp. 139-142. https://doi.org/10.1145/2857491.2857520

[82] Saccade, Wikipedia. https://en.wikipedia.org/wiki/Saccade, accessed on Oct. 10, 2024.

[83] Lobão-Neto, R., Brilhault, A., Neuenschwander, S., Rios, R. (2022). Real-time identification of eye fixations and saccades using radial basis function networks and Markov chains. Pattern Recognition Letters, 162: 63-70. https://doi.org/10.1016/j.patrec.2022.08.013

[84] Fuhl, W., Kasneci, G., Kasneci, E. (2021). Teyed: Over 20 million real-world eye images with pupil, eyelid, and iris 2d and 3d segmentations, 2d and 3d landmarks, 3d eyeball, gaze vector, and eye movement types. In 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Bari, Italy, pp. 367-375. https://doi.org/10.1109/ISMAR52148.2021.00053

[85] Maquiling, V., Byrne, S.A., Niehorster, D.C., Carminati, M., Kasneci, E. (2025). Zero-shot pupil segmentation with SAM 2: A case study of over 14 million images. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 8(2): 1-16. https://doi.org/10.1145/3729409

[86] ReduceLrOnPlateau, PyTorch. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html, accessed on Oct. 10, 2024.

[87] Accurate CNN-based Pupil Segmentation with an Ellipse Fit Error Regularization Term. (2021). Github, Cuneytakinlar, UNet Implementation. https://github.com/cuneytakinlar/pupil-ellipse-fit-error/blob/master/unet/unet_model.py, accessed on Oct. 10, 2024.

[88] 2D-UNet-Pytorch. (2020). Github, ProfessorHuang, UNet++ Implementation. https://github.com/ProfessorHuang/2D-UNet-Pytorch/blob/main/models/nested_unet.py, accessed on Oct. 10, 2024.

[89] TransUNet. (2021). Github, Beckschen, TransUNet Implementation. https://github.com/Beckschen/TransUNet/blob/main/networks/vit_seg_modeling.py, accessed on Oct. 10, 2024.

[90] Torchvision. (2023). Github, PyTorch, DeepLabv3 Implementation. https://github.com/pytorch/vision/tree/main/references/segmentation, accessed on Oct. 10, 2024.

[91] PaddleSeg (2023) Github, PP-MobileSeg Implementation. https://github.com/PaddlePaddle/PaddleSeg/blob/release/2.8/paddleseg/models/pp_mobileseg.py, accessed on Oct. 10, 2024.

[92] Devil is in Channels: Contrastive Single Domain Generalization for Medical Image Segmentation. (2023). Github, ShishuaiHu, CCSDG Implementation. https://github.com/ShishuaiHu/CCSDG/blob/master/ccsdg/models/unet_ccsdg.py, accessed on Oct. 10, 2024.

[93] EGE-UNet (2023) Github, JCruan519, EGEUnet. https://github.com/JCruan519/EGE-UNet/blob/main/models/egeunet.py, accessed on Oct. 10, 2024.

[94] Amplitude-based traditional method source published by Wan et al. (2020) Github, https://github.com/zhwan934940943/CPRD/tree/master, accessed on Oct. 10, 2024.

[95] LEyes (2023) Github, LEyes implementation https://github.com/dcnieho/Byrneetal_LEyes/tree/master, accessed on Oct. 10, 2024.

[96] Garbin, S.J., Shen, Y., Schuetz, I., Cavin, R., Hughes, G., Talathi, S.S. (2019). Openeds: Open eye dataset. arXiv preprint arXiv:1905.03702. https://doi.org/10.48550/arXiv.1905.03702

[97] Palmero, C., Sharma, A., Behrendt, K., Krishnakumar, K., Komogortsev, O.V., Talathi, S.S. (2020). Openeds2020: Open eyes dataset. arXiv preprint arXiv:2005.03876. https://doi.org/10.48550/arXiv.2005.03876